AD-772 490

THE EFFECTS OF MESO-SCALE AND SMALL-SCALE INTERACTIONS ON GLOBAL CLIMATE

Systems, Science and Software

Prepared for:

Defense Advanced Research Projects Agency

7 January 1974

SYSTEMS, SCIENCE AND SOFTWARE

SSS-R-74-2023

THE EFFECTS OF MESO-SCALE AND SMALL-SCALE
INTERACTIONS ON GLOBAL CLIMATE

Semiannual Technical Report
for Period
17 April 1973 through 15 October 1973

7 January 1974

i

TABLE OF CONTENTS

TABLE OF CONTENTS, contd.

INTRODUCTION

ATMOSPHERIC RADIATION

The funds for the atmospheric radiation portion of the contract were exhausted as of the previous report; however, work has continued with other support. A summary of the progress made follows.

Major effort was directed towards an extensive study of solar radiation fluxes in the presence of typical summer Arctic stratus clouds. A paper on this subject was delivered by Dr. Warren J. Wiscombe at the 24th Alaskan Science Conference in Fairbanks, Alaska (August 15-17, 1973). A considerably extended and generalized version of this paper has been submitted for publication in the proceedings of that conference.

Massive code modifications were made in ATRAD (the atmospheric radiation computer model) which:

(1) introduced a specular reflection option (important for calculations involving smooth or slightly roughened sea surfaces);

(2) allowed the use of a grey-body top boundary with arbitrary temperature and emissivity, so that, for example, long-wave (IR) calculations need not include

the whole atmosphere but may be terminated at a cloud base;

(3)   streamlined the basic Grant-Hunt algorithm computation significantly;

(4)   allowed the code to cycle over any one of the following parameters -- sun angle, surface reflectivity, surface temperature, top-boundary temperature.

The cycling feature in (4) is a tremendous aid in making extensive parameter studies such as were required for the Arctic stratus problem, and was achieved with virtually no increase in computing time over previous calculations involving a single parameter. This was possible because most of the calculations in ATRAD do not depend on a particular parameter, and thus it was possible to cycle only those parts of the calculation depending on the parameter in question. Since any one of four parameters may be cycled, the code modifications were necessarily complex and time-consuming.

Finally, a seminar "Comparisons of a Detailed Radiation Model with the Radiation Subroutine of the Mintz-Arakawa General Circulation Model" was prepared and presented as part of the weekly seminar series of the UCLA Meteorology Department.

 OROGRAPHIC EFFECTS ON GLOBAL CLIMATE

The results reported below pertain to the continuation of numerical investigations of meso-scale momentum transfer in the atmosphere. Investigations can be separated into two parts: the transient and the steady-state.

The steady-state part has culminated in the development of two- and three-spatial dimensional codes based on the formulation of Bretherton.[1] Additional analysis of the Bretherton equations is found in Section 2, and a 3-D study of the wave drag over the Sierra Nevada range of California is described in Section 3. The steady-state computer code is divided into three parts to more efficiently conduct parameterization studies. This work is described in Section 4.

Transient phenomena have been investigated with the 2-D code HAIFA, and the 3-D code STUFF. The latter has been optimized in terms of speed of execution and of fast storage requirements. A complete description of the code optimization, along with a discussion of the numerical techniques employed by STUFF, is given in Part II of this report.

During the next contract period major emphasis will be placed on application of the transient and steady-state codes to the development of heuristic expressions characterizing meso-scale momentum transfer processes induced by global topography.

SYSTEMS, SCIENCE AND SOFTWARE

PART I

DEVELOPMENT OF STEADY-STATE MODELING

4

1. INTRODUCTION

By the close of the last contract period, the development of a 3-D, linear, steady-state code based on the Bretherton[1] formulation had been completed. An outline of the code was also presented. Since then, several subtle points of Bretherton's paper have been analyzed. A discussion of these is presented in Section 2. Section 3 presents the results of a 3-D steady-state study of the Sierra Nevada-Owen's Valley region of north-central California where lee-wave measurements have been carried out.

Analysis of the 3-D code during this contract period has shown that it might be significantly optimized by reorganization and recoding. The results of this task are presented in Section 4. A discussion of the numerical method is included. Finally, a discussion of plans for future work is presented in Section 5.

2. ADDITIONAL ANALYSIS OF THE BRETHERTON FORMULATION

The Eq. (54) of Bretherton,[1] on which the 3-D steady-state evaluation of orographic gravity wave drag is based, was presented in his original paper substantially without derivation. Subsequently, several of the steps in the derivation were presented in a recent progress report[2] on calculations usin the Bretherton formulation. Several points in the Bretherton paper, however, which were not examined at that time, subsequently have been found to contain subtlties. In addition, errors in the text have been recognized and corrected. Consequently, we briefly record below the salient arguments required to complete the Bretherton formulation.

2.1 CHOICE OF VERTICAL VELOCITY AT UPPER BOUNDARY

The solution of the vertical velocity equation requires boundary conditions to be imposed both at the ground and at the top of the atmosphere. In order to perform a marching calculation, a specific value of the complex vertical velocity is assigned at the top of the atmosphere. The resulting wave drag is independent of this value, a result which we wish to demonstrate below. In order to do so, we evaluate the quantity F containing all of the vertical velocity terms. According to Bretherton, we must consider the two cases for which different boundary conditions at the top of the atmosphere $z=H$ are prescribed.

Before considering these two cases, however, we develop a general solution of the Scorer equation with which we can more easily consider the boundary conditions. The real and imaginary parts of the vertical velocity each obey the same linear second order equation

$$\frac{d^2u}{dz^2} + (\ell^2 - \kappa^2)u = 0 \quad . \tag{1}$$

A general solution of this equation contains two coefficients

$$u = a_1u_1 + a_2u_2 \tag{2}$$

where u_1 and u_2 are linearly independent. We choose the functions u_1 and u_2 to be solutions of Eq. (1) having the following boundary conditions:

$$u_1(H) = 1 \quad , \qquad \left.\frac{du_1}{dz}\right)_H = 0 \quad ,$$

$$\tag{3}$$

$$u_2(H) = 0 \quad , \qquad \left.\frac{du_2}{dz}\right)_H = 1 \quad .$$

As mentioned above, the real and imaginary parts of $\hat{w}$, the vertical component of the perturbation velocity, each have the form of Eq. (2).

We now discuss the simpler of the two cases in which $\kappa^2 > \ell^2(H)$, for which we have trapped wave solutions and the contributions to the Reynold's stress are discrete (as given by Eq. (61) of Bretherton). The boundary condition at the top of the atmosphere $z=H$ in this case, is Eq. (50a) of Bretherton:

$$\frac{d\hat{w}}{dz} = - \sqrt{\kappa^2 - \ell^2(H)}\,\hat{w} \quad .$$

Assuming an initial value of $\hat{w}$ at $z=H$ of $\hat{w}(H) = u_0 + iv_0$, the initial values of the derivatives are

$$\left.\frac{dw_R}{dz}\right)_H = - \sqrt{\kappa^2 - \ell^2(H)}\,u_0 \quad ,$$

$$\left.\frac{dw_I}{dz}\right)_H = - \sqrt{\kappa^2 - \ell^2(H)}\,v_0 \quad .$$

We can now determine the coefficients a_1 and a_2 in Eq. (2):

$$w_R = u_0(u_1 - \sqrt{\kappa^2 - \ell^2(H)}\,u_2) \quad ,$$

$$w_I = v_0(u_1 - \sqrt{\kappa^2 - \ell^2(H)}\,u_2) \quad . \tag{4}$$

Thus, we find that the real part of the velocity w_R and the imaginary part w_I are proportional to each other at every altitude z . Consequently, it is only necessary to perform one integration of the Scorer equation to evaluate either one of them.

It is now possible to construct the term containing all of the velocity dependence of the Reynold's stress, given by Bretherton's Eq. (60);

$$\int F(\kappa)\,d\kappa = \frac{\pi}{2\kappa}\,\frac{\left|\left.\frac{d\hat{w}}{dz}\right)_0\right|^2}{\int_0^H |\hat{w}|^2\,dz} = \frac{\pi}{2\kappa}\,\frac{\left(\frac{dw_R(0)}{dz}\right)^2 + \left(\frac{dw_I(0)}{dz}\right)^2}{\int_0^H (w_R^2 + w_I^2)\,dz} \quad .$$

8

In terms of the expressions of Eq. (4), we obtain

$$\int F(\kappa)\,d\kappa = \frac{\pi}{2\kappa}\,\frac{\left(\frac{du_1}{dz} - \sqrt{\kappa^2-\ell^2(H)}\,\frac{du_2}{dz}\right)^2_{z=0}}{\int_0^H (u_1 - \sqrt{\kappa^2-\ell^2(H)}\,u_2)^2\,dz} \quad . \tag{5}$$

The salient feature of Eq. (5) is that the values of u_o and v_o do not appear; the trapped wave contribution to the Reynold's stress is independent of the assumed velocity amplitude at $z=H$.

For the case of waves which leak into the stratosphere, corresponding to $\kappa^2 \le \ell^2(H)$, the boundary condition (given by Eq. (50b) of Bretherton) is

$$\frac{d\hat{w}}{dz} = + i\sqrt{\ell^2(H)-\kappa^2}\,\mathrm{sgn}(U_n)\hat{w} \quad .$$

Substituting the assumed boundary values at $z=H$ the derivative condition becomes

$$\frac{dw_R}{dz}\bigg)_H = -\sqrt{\ell^2(H)-\kappa^2}\,\mathrm{sgn}(U_n)v_o \quad ,$$

$$\frac{dw_I}{dz}\bigg)_H = \sqrt{\ell^2(H)-\kappa^2}\,\mathrm{sgn}(U_n)u_o \quad .$$

The general solutions satisfying these boundary conditions are

$$w_R = u_1 u_o - \sqrt{\ell^2(H)-\kappa^2}\,\mathrm{sgn}(U_n)u_2 v_o \quad ,$$

$$w_I = u_1 v_o + \sqrt{\ell^2(H)-\kappa^2}\,\mathrm{sgn}(U_n)u_2 u_o \quad . \tag{6}$$

9

In contrast to the first case, these two solutions are linearly independent and require that two integrations of the Scorer Equation be performed.

In order to evaluate the term containing all of the dependence on the vertical velocity in the Reynold's stress, we compute the quantity F (see Bretherton's Eq. (52)):

$$F = \frac{1}{2i} \left\{ \frac{d\hat{w}}{dz}\hat{w}^* - \frac{d\hat{w}^*}{dz}\hat{w} \right\} \Big/ \hat{w}(0)\hat{w}^*(0) \quad , \qquad (7)$$

where w^* denotes the complex conjugate of w . The numerator of F is independent of altitude and can be evaluated at z=H from the boundary values given above. The result is

$$\frac{1}{2i} \left\{ \quad \right\} = \sqrt{\ell^2(H)-\kappa^2} \; \text{sgn}(U_n) \; \hat{w}(H) \; \hat{w}^*(H)$$

$$= \sqrt{\ell^2(H)-\kappa^2} \; \text{sgn}(U_n) \, (u_o^2+v_o^2) \quad .$$

Using this result, and forming the denominator of Eq. (7) from the solutions of Eq. (6) evaluated at z=0 , we obtain for F

$$F = \frac{\sqrt{\ell^2(H)-\kappa^2} \; \text{sgn}(U_n)}{u_1^2(0) + (\ell^2(H)-\kappa^2)u_2^2(0)} \quad . \qquad (8)$$

This expression is also independent of the assumed boundary values u_o and v_o . Consequently, we have shown that the Reynold's stress does not depend on the chosen values of vertical velocity at the top of the atmosphere.

2.2   REDUCTION OF DOMAIN OF INTEGRATION

This result, which reduces the amount of calculation in forming the Reynold's stress components by a factor 2, is incorrectly justified by Bretherton. In fact, as will be shown below, the simplification results from the integrand being the same at the wave numbers (k, ℓ) and $(-k, -\ell)$ representing a reflection in the origin of wave number space, rather than as stated by Bretherton that (κ, ϕ) is the same as $(\kappa, -\phi)$.

Several factors enter in the Reynold's stress integrand; we examine the behavior of each of these when the transformation $(-k, -\ell) \rightarrow (k, \ell)$ is carried out. First, we examine the differential equation for the vertical velocity and its boundary conditions.

$$\kappa^2(-k, -\ell) = k^2 + \ell^2 = \kappa^2(k, \ell)$$

$$U_n(-k, -\ell) = \frac{-1}{\kappa}(Uk + V\ell) = -U_n(k, \ell) \tag{9}$$

$$\ell^2(-k, -\ell) \equiv \frac{N^2}{U_n^2} - \frac{d^2 U_n}{U_n \, dz^2} = \ell^2(k, \ell) \quad .$$

i.e., the Scorer parameter, being an even function of U_n, is invarient under the transformation. Consequently, the Scorer equation and its elementary solutions, depending only on $\ell^2 - \kappa^2$, are also invariant. Using this result, which establishes that u_1 and u_2 of Eq. (3) are invariant, and considering the transformation properties of Eq. (8) for F, we find that

$$F(-k, -\ell) = -F(k, \ell) \tag{10}$$

We now consider the topographic factor of the Reynold's stress given by Eq. (47) of Bretherton.

$$\hat{h}(-k,-\ell) = \frac{1}{4\pi} \int_0^X \int_0^Y h(x,y) \, e^{-i(-kx-\ell y)} \, dxdy$$

$$= \hat{h}*(k,\ell) \quad .$$

Forming the spectrum function $A = \frac{4\pi^2}{XY} \hat{h} \hat{h}*$, we obtain the transformation property

$$A(-k,-\ell) = \frac{4\pi^2}{XY} \hat{h}(-k,-\ell) \, \hat{h}*(-k,-\ell) = A(k,\ell) \quad .$$

These quantities can be combined to form the integrands of the two horizontal components of the Reynold's stress:

$$\begin{Bmatrix} \rho uw \\ \rho vw \end{Bmatrix} = \rho_o U_n^2(0) \; \kappa^2 AF \begin{Bmatrix} \cos\phi \\ \sin\phi \end{Bmatrix} = \rho_o U_n^2(0) \; \kappa AF \begin{Bmatrix} k \\ \ell \end{Bmatrix} \quad .$$

The transformation $(-k,-\ell) \to (k,\ell)$ is seen to leave the integrand invarient:

$$\rho uw(-k,-\ell) = \rho uw(k,\ell)$$

$$\rho vw(-k,-\ell) = \rho vw(k,\ell) \quad .$$

Consequently, the result given in Bretherton's Eq. (54) is confirmed; namely that

$$\int_0^{2\pi} d\phi = 2 \int_0^\pi d\phi \; .$$

3. COMPARISON OF TRANSIENT AND STEADY-STATE WAVE DRAG

3.1 INTRODUCTION

At the close of the last contract period, two investigations of wave drag phenomena associated with the orography of the Sierra Nevada had been completed. These were the 2-D transient study presented in the preceding report,[2] and the 2-D linear steady-state study discussed in another report.[3] In the transient study, atmospheric flow over the Sierra Nevadas was simulated, resulting in the time-dependence of wave drag values. Additionally, the effects of moisture on wave drag were evaluated. A steady-state calculation was also performed, from which a small-scale parameterization using some of the important dynamic quantities was developed. The results of this calculation, where applicable, were compared to the transient results.

3.2 THREE-DIMENSIONAL STEADY-STATE STUDY

After the completion of the 3-D steady-state code, two test problems were run. The first investigated a generalization of the triangular mountain of the previous report[2] by calculating a triangular ridge 128 km in length. The second calculation was applied to the actual Sierra Nevada-Owen's Valley topography described below.

The Sierra Nevada is a single unbroken range approximately 400 miles in length and 50 to 80 miles wide. At the·

Owen's Valley we find the main crest running nearly North-South. Elevations along the crest are around 12,000 ft. with numerous peaks rising above 14,000 ft. The eastern scarp is abrupt and quite straight. Owen's Valley is to the east at an average elevation of 4,000 ft. and of a nearly uniform width of approximately 15 miles. The east wall of Owen's Valley is a fault block range called the Inyo Mountains to the south and the White Mountains to the north. The elevation of the Inyo Mountains is 7,000-11,000 ft. while the White Mountains rival the Sierras in height.

Figures 1 and 2 show the topography of the Owen's Valley region. Figure 1 is a contour map of the data used in the second 3-D calculation. Figure 2 shows three east-west transects at the indicated latitudes. The asterisk indicates the position of Owen's Valley. Both figures were made from data derived from the 5' × 5' topography tapes described in Section 4.

The steady-state runs utilized the wind and temperature profile data obtained from the Merced weather station and employed in the previous studies.[2,3] The calculations indicate a principal trapped wave having a wavelength of approximately 19 km. This agrees well with the value observed in the Owen's Valley under the same meteorological conditions of 18-20 km. The wave drag results are shown superimposed on the transient HAIFA results in Figure 3. Two-dimensional steady-state calculation results utilizing the east-west transects of Figure 2 are also presented. As can be seen, the results of the 3-D runs agree quite closely, revealing a possible insensitivity to the topography.

The 2-D runs, however, differ by as much as a factor of 3. Since the real topography calculations and idealized topography 3-D calculations agree closely, one may assume that

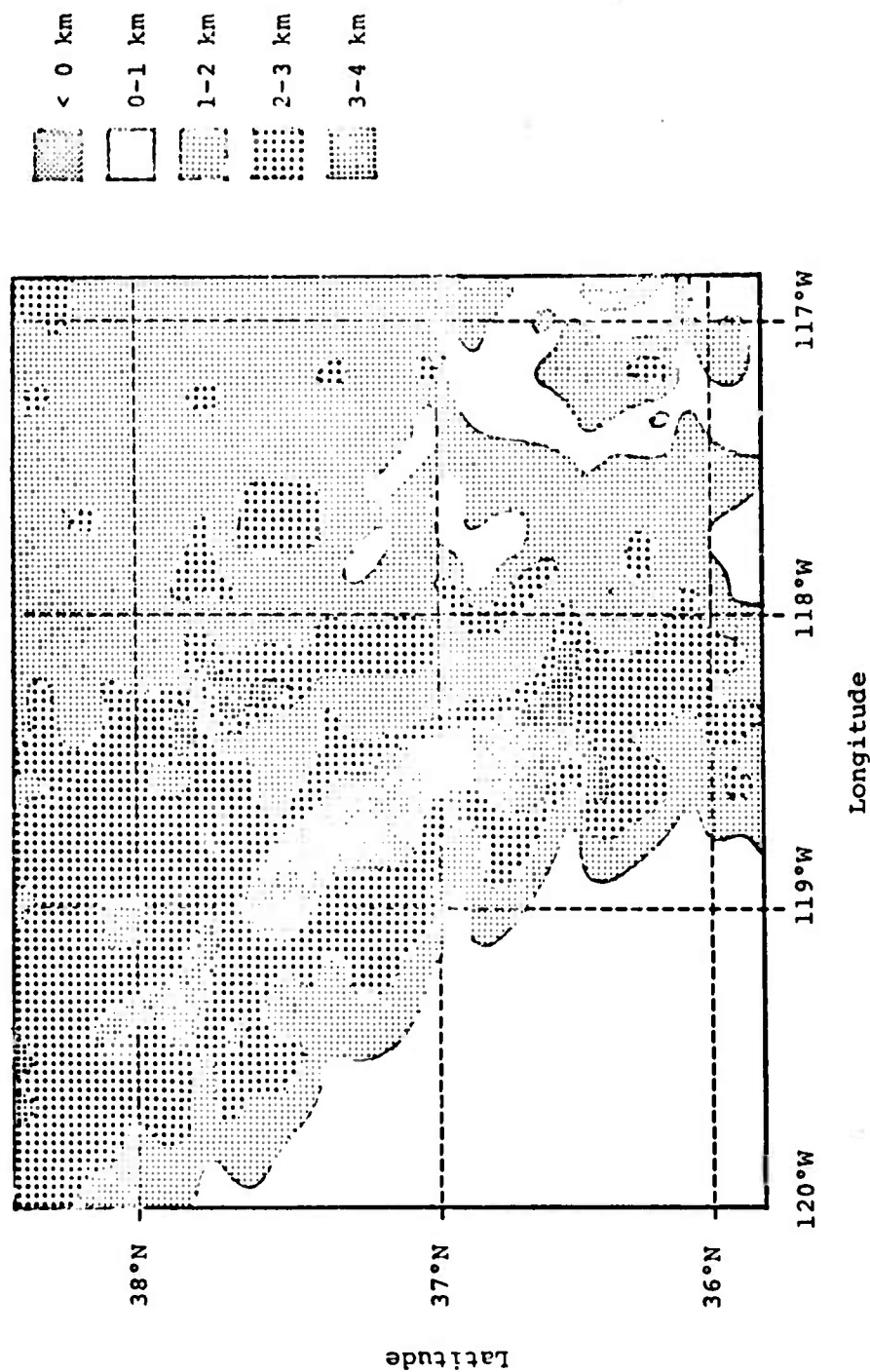.Figure 1   Contour map of the Sierra Nevada-Owen's Valley region showing contours at 1 km intervals. Data are obtained from Defense Mapping Agency 5'×5' topography tables.

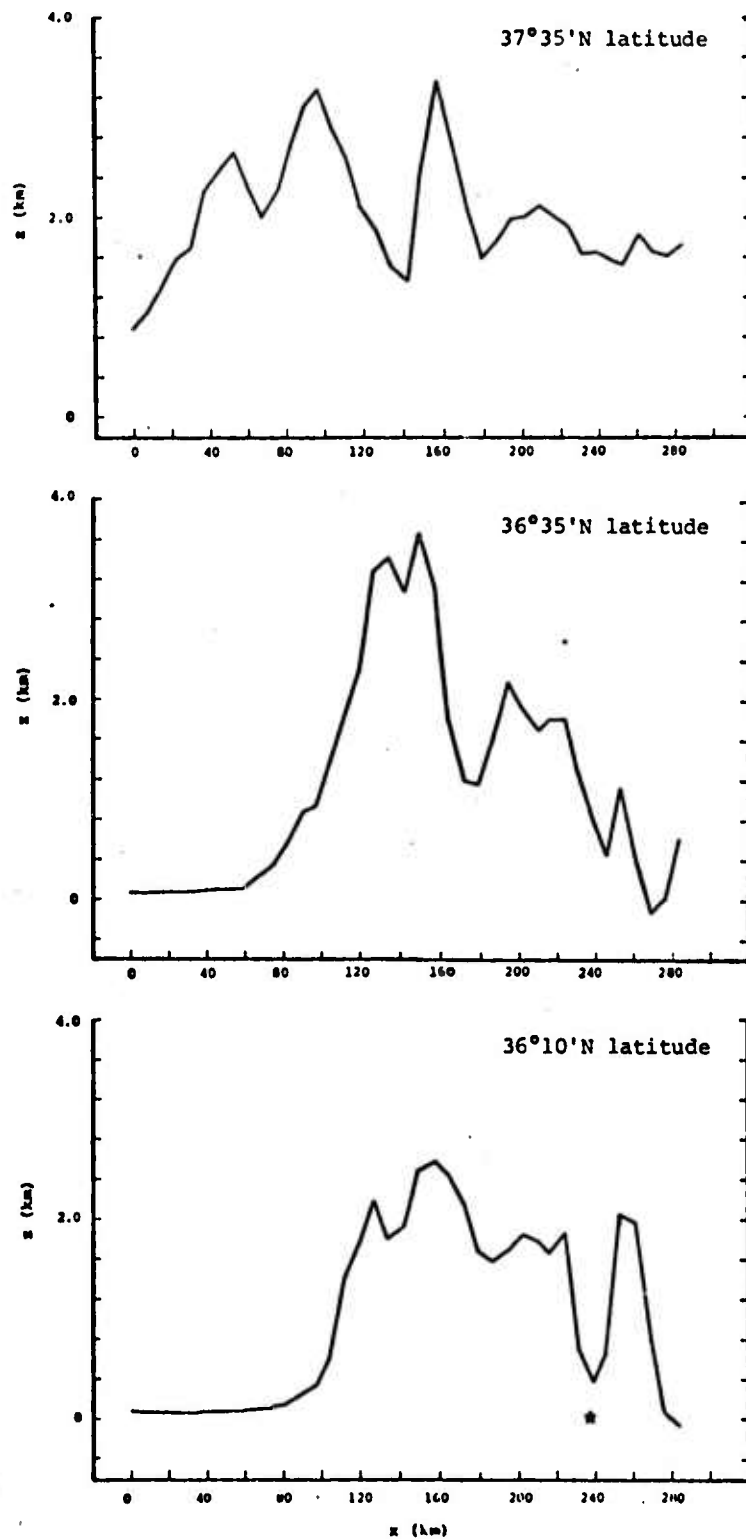Figure 2   2-D cross-sections of Sierra topography from 120°W
            through 116°50'W longitude. Asterisk indicates
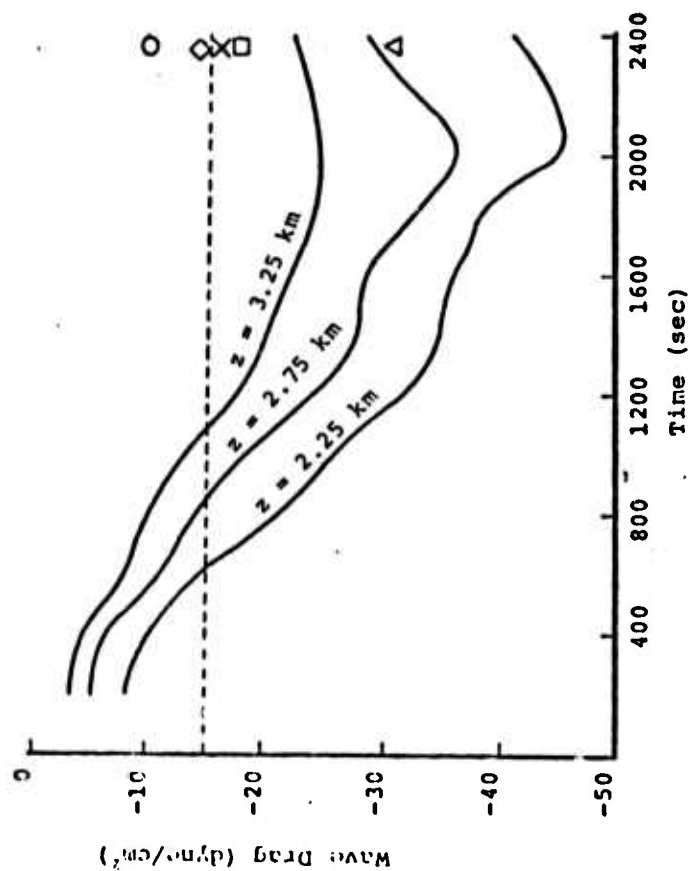   Owen's Valley.

Figure 3   Vertical flux of horizontal momentum for HAIFA, compared with steady-state calculations.

the idealized topography is a good characterization of the real topography. The differences between the 2-D and 3-D results can thus be attributed to the sensitivity of Reynold's stress to the topographic variability exhibited in Figure 2.

These results provide some indirect evidence of the accuracy with which topographic data must be represented. The question of sensitivity to the representation of topographic data will be examined in greater detail during the next contract period.

It is of interest to examine the wavenumber dependence of the atmospheric factor F (the spectral Reynold's stress corresponding to unit topographic spectrum function). This quantity can indicate to us what spectral regions are most significant in terms of both the continuous and trapped wave Reynold's stress contributions. We examine below this quantity for the atmospheric parameters of the Merced problem. The distribution of the trapped waves is seen in Figure 4. The locus for a particular trapped wave is generated by examining the solution of the Scorer equation for each of the plotted points. The locus is then constructed by connecting points whose solutions have identical numbers of nodes. Current $\Delta\phi, \Delta\kappa$ values allow the identification of three trapped waves. These waves are modest contributors to the drag for the Sierra Nevada problem. Apparently, other waves which are not adequately resolved in the figure can be substantial contributors as well. Some of the individual points not associated with an established locus have been found to contribute more than those found on the loci. It appears as though integrations having higher resolution in $\Delta\kappa, \Delta\phi$ are required to determine these trapped wave contributions more fully.

One worrisome feature of the analytic expression for the trapped wave contribution to the drag is that waves whose
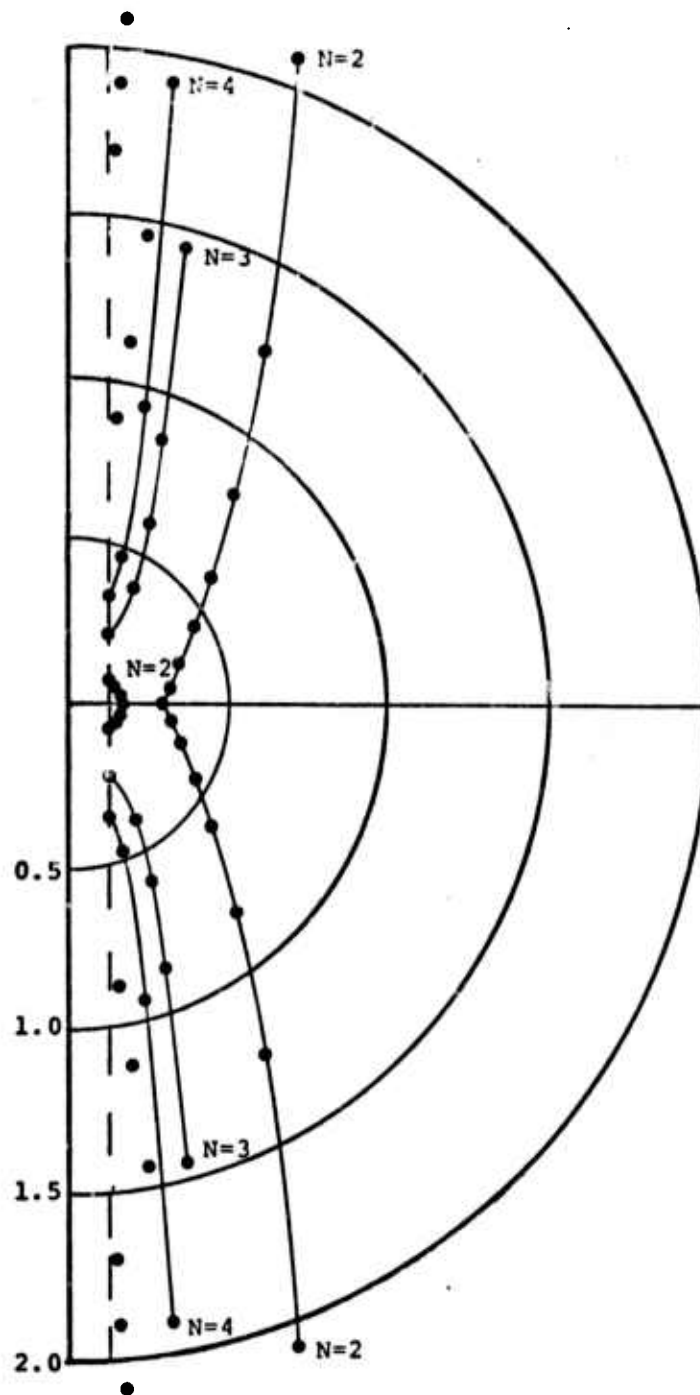
18

Figure 4   Trapped wave distribution in κ, ϕ space. Points not falling on curves represent contributors to unresolved trapped waves. N is the number of nodes.

loci are essentially radial in κ, ϕ space may be inaccurately integrated in the wave drag calculation. This is due to the fact that an increment of $\Delta\phi$ may miss the trapped wave partially or even completely. Some of the loci of the waves of Figure 4 exhibit quite nearly radial behavior and consequently some of the contribution may be lost. A change to integrating over k, ℓ instead of $\kappa \phi$ as independent variables might alleviate this since the spatial resolution would be more easily controlled at greater distances from the origin.

Figure 5 is a contour plot of the continuous contribution to the F distribution for the Sierra Nevada problem. Both the continuous and the trapped waves exhibit symmetry about $\phi=0°$ in this case. The reason for this can be seen in the symmetry of the Scorer parameter and the boundary condition, which in turn implies symmetry in the solutions of the Scorer equation. For the Sierra problem U_n is given by

$$U_n = U(z) \cos\phi \quad .$$

Since the symmetry of the Scorer parameter depends on the symmetry properties of U_n, we see that the Scorer parameter will be symmetric about $\phi=0°$.

The F values in Figure 5 remain large even for κ values quite close to $\ell(H)$. In fact, for some ϕ, the F values actually increase until $\kappa = \ell(H)$. This behavior can be explained by examining the expression for F :

$$F = 2\sqrt{\ell^2(H)-\kappa^2} \; \text{sgn}(U_n)/w^*(0)w(0) \quad .$$

One sees that the κ-dependence of F is predominantly controlled by the denominator for $\kappa \ll \ell(H)$. For values of κ near $\ell(H)$, however, the behavior of F can be affected by
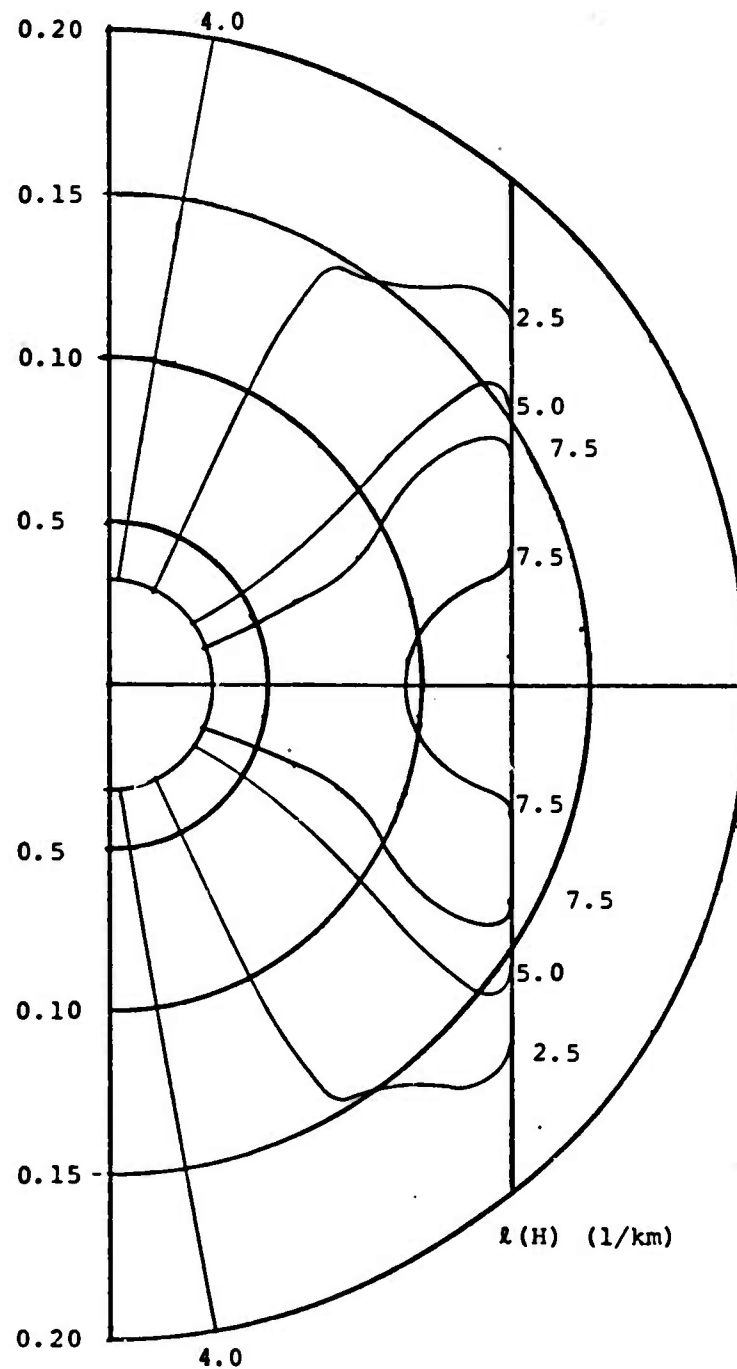
Figure 5   F   distribution in κ, ϕ space.

both numerator and denominator. If the denominator is not singular near $\kappa = \ell(H)$ F will decay monotonically to zero as κ approaches $\ell(H)$. However, if the denominator has a zero near $\ell(H)$, a resonant behavior can occur in F.

This behavior corresponds to the existence of a trapped wave near $\ell(H)$. While this situation was not expected to occur very often, the Sierra Nevada calculations reveal several such cases as shown in Figure 6. The methods used in the numerical integration currently do not take account of this occurrence accurately; an improved integration routine will be developed to deal with this case.

Apparently, this near coincidence of a resonance with the locus of $\ell(H)$ did not take place in the example calculated by Bretherton. As a consequence, he did not find it necessary to take special care with the integration routine in this region. As additional examples are investigated it is to be expected that other special cases will be found which also will require special consideration.
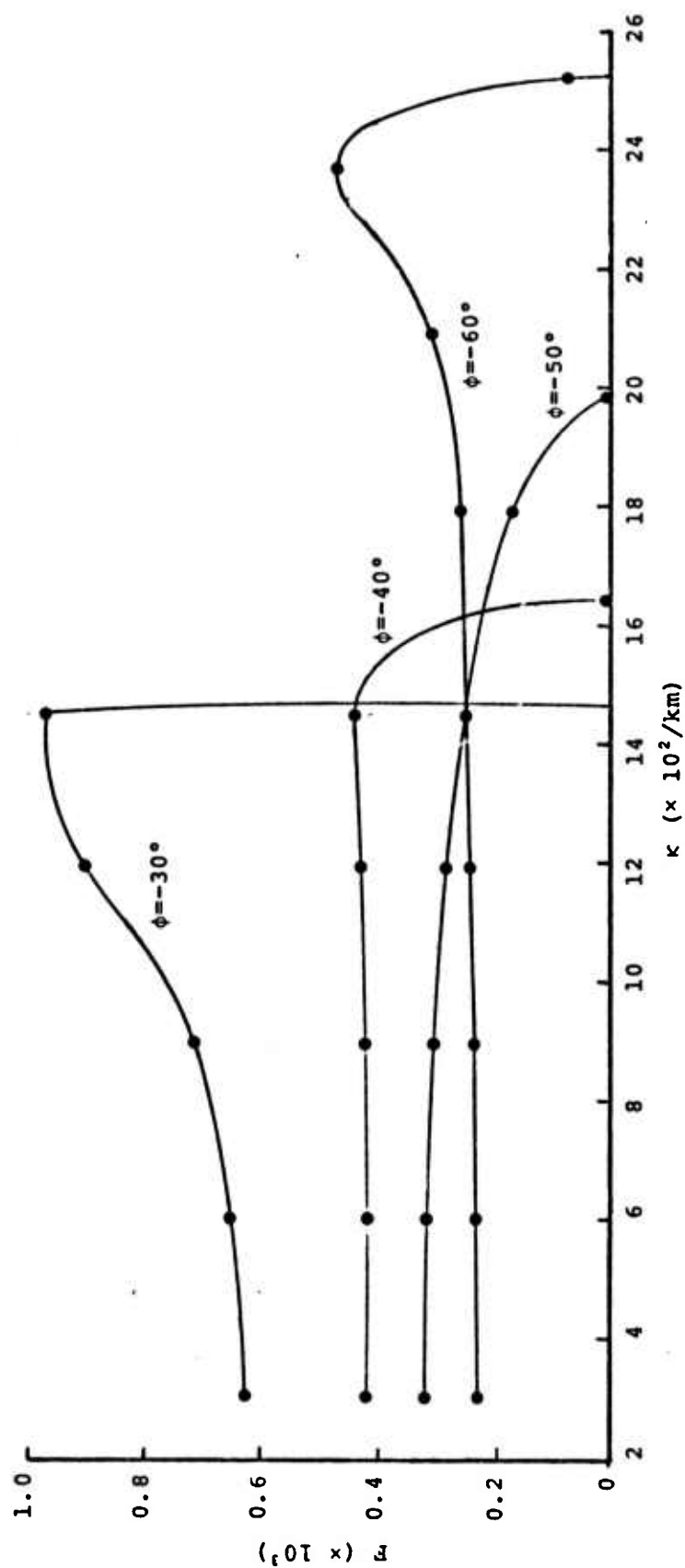
Figure 6   F(κ) distribution for various φ angles.

4. FURTHER CODE DEVELOPMENTS

4.1 INTRODUCTION

When the steady-state codes became operational, various test problems were run to determine their response to different problem configurations. The comparison of the results of a 3-D run and the previously reported[2] HAIFA study of the Sierra Nevada problem were presented in Section 3. They indicate that similar Reynold stress values are obtained by these two markedly different codes.

On completion of this comparison, work was begun to improve the efficiency of the steady-state code, preparatory to a large number of survey calculations. This program is discussed more fully in Section 5. In summary, the code is segmented such that the spectrum function, F factor and stress integrals are calculated independently of each other, but exchange data through files established on mass storage devices. This procedure makes efficient use of fast core storage and decreases computer costs. At the same time, much greater efficiency is achieved in performing a matrix of calculations. Since this modulization has resulted in major design changes in the codes, a description of the new routines follows. The topography data files have also undergone major modification and streamlining. These are also discussed below.

4.2   TOPOGRAPHY

Topography data were obtained from the Defense Mapping Agency (DMA) and are contained on two sets of magnetic tapes. The first set contains elevations area-averaged over 30'×30' rectangles for the entire globe. The second set contains elevations area-averaged over 5'×5' rectangles for most of North America and Europe.

An examination of these tapes indicated that a restructuring of the data format would allow much faster access to particular pieces of information. As a result, the reading of all available 30' data now requires 30 seconds of CPU time as compared to 600 seconds previously.

The new tapes, resulting from the above restructuring, are arranged in a common format for both the 30' and 5' data sets. The 30' set requires two tapes, one for the northern hemisphere and one for the southern hemisphere. The 5' set consists of one tape covering the northern hemisphere. The data are written in logical records having 1444 word length for 30' data and 4324 words for 5' data. Each logical record represents topographical data at a particular latitude for a full 360° of longitude. These records are written on the tapes sequentially starting at the northernmost latitude of each hemisphere. At the head of each logical record are written the starting latitude and longitude of the strip, where longitude is measured east of Greenwich. Associated with each data point is a code number which contains information about the nature of the topography; i.e., whether sea bed, lake, all land, etc. This coded information is the same as that in the DMA description of their tapes. In addition to the code data, however, another indicator has been added to denote missing data. It was found that occasional gaps in data occurred on

25

the DMA tapes. Since the Fourier transform routine requires
complete data, a flag is set whenever the missing topography
is encountered and the run is terminated. Usually, this re-
quires obtaining the data from other sources and inserting it
into the data tape. The case can then be rerun in the usual
manner.

4.3   THE SPECTRUM FUNCTION CODE

4.3.1   The Equations

The spectrum function code performs the primary task
of determining a spectrum function distribution in wavenumber
space associated with a Fourier representation of the hori-
zontal dependence of the topography. Bretherton[1] defines a
spectrum function, A;

$$A(k,l) = \frac{4\pi^2}{XY} \hat{h}* \hat{h} \tag{1}$$

where X and Y represent the spatial extent of the topo-
graphic region in the east-west and north-south directions. $\hat{h}$
is the Fourier transform of the surface height and the asterisk
indicates conjugation. The Fourier transform is defined as

$$\hat{h}(k,l) = \frac{1}{4\pi^2} \int_o^X \int_o^Y h(x,y) e^{-i(kx+ly)} dxdy \tag{2}$$

These two equations are solved in Module 1.

4.3.2   Numerical Method

The approach taken in Module 1 is as follows: First,
surface height data are obtained from the appropriate data
tape. Next, these data are Fourier transformed to obtain the
spectrum. Finally, the spectrum function distribution is cal-
culated according to Eq. (1).

26

The topography data are composed of m×n data "points" falling within the rectangle specified by the user at problem generation time. It must be remembered that each "point" is, in fact, an average value to be associated with a small rectangle about the point in question. Thus, the surface height may be thought of as a step-function representation of the topography. A sample 5×4 grid shown in Figure 7 demonstrates this more clearly. Each of the numbers appearing within the cells represents the average height within the cell.

The nature of the data representation has implications for the Fourier transform. First, the discrete data can be represented by a discrete number of spectral components. Second, the resolution in spectral space is a function of the resolution in real space. As mentioned in a previous report, [2] this lack of spectral resolution can affect the wave drag calculation through rendering A uncertain. Particularly at trapped wavenumbers this uncertainty results in a corresponding uncertainty in the trapped wave drag contribution. The use of 5' data allows resolution of wavelengths of the order of 10 km, which are comparable with the wavelengths of the predominant trapped waves. We propose to examine the desirability of using even more highly resolved data.

The fact that the topography can be considered discrete is fortunate from a calculational point of view, however. Calculation of the Fourier transform requires a large amount of time due to the large number of trigonometric evaluations and manipulations of the data. The fast Fourier transform (FFT), which is an algorithm to optimize the direct calculation of the Fourier transform, is used to calculate the topography spectrum function. The algorithm requires that the function be defined at a discrete set of points spaced at equal intervals in each dimension. A FORTRAN subroutine based on the Cooley Tukey
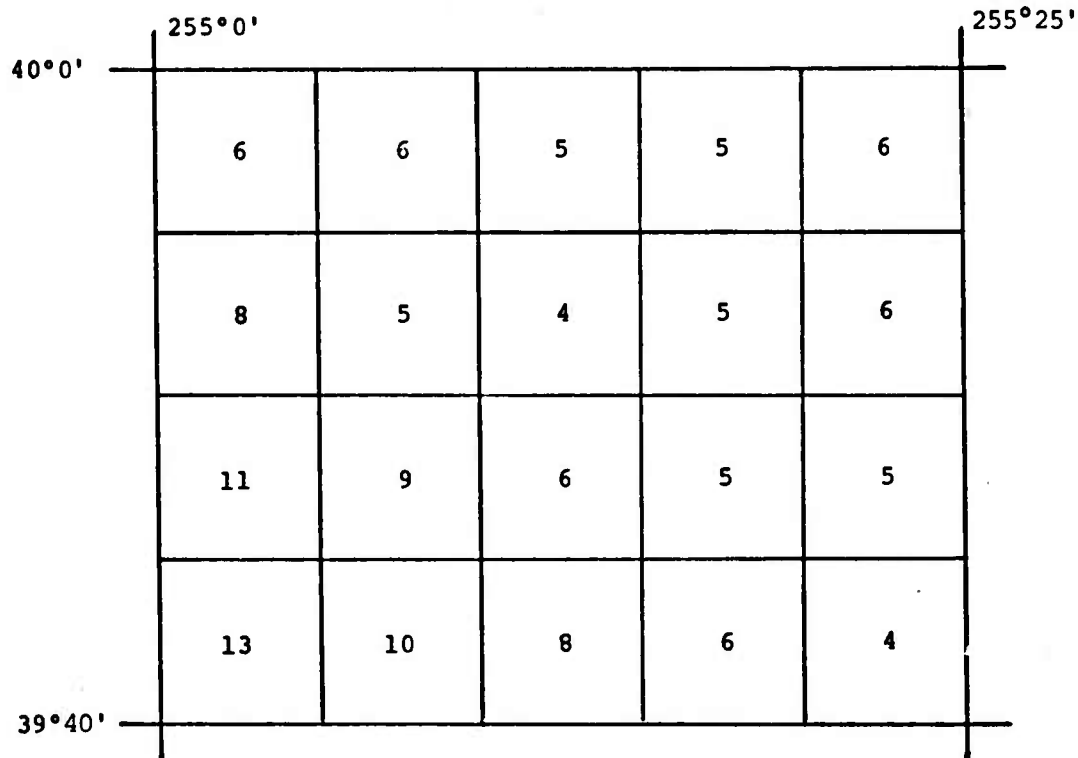
Figure 7   A hypothetical topographical grid extending 25'×20'.
           The numbers represent mean values for the cell. Grid
           resolution is 5 minutes.

28

algorithm was obtained from the University of California at San Diego and has been used satisfactorily in the applications described below.

The finite Fourier transform requires some adaptation before it is applicable to the topography data. This can be seen from the finite difference expression for the Fourier integral which is given by

$$\hat{h}(k,l) = \frac{1}{4\pi^2} \sum_{s=0}^{m-1} \sum_{u=0}^{n-1} h(s\Delta x, u\Delta y) e^{-i(ks\Delta x + lu\Delta y)} \Delta x \Delta y \tag{3}$$

The FFT routine calculates

$$\tilde{a}_{rt} = \frac{1}{mn} \sum_{s=0}^{m-1} \sum_{u=0}^{n-1} a_{su} e^{-2\pi i \left(\frac{rs}{m} + \frac{tu}{n}\right)} \tag{4}$$

We see then that if we identify a_{su} with the height function as follows

$$a_{su} = \frac{m\Delta x \; n\Delta y}{4\pi^2} h(s\Delta x, u\Delta y) \tag{5}$$

and define

$$k = \frac{2\pi r}{m\Delta x} \qquad\qquad 0 \le r < \frac{m}{2} \tag{6}$$

$$k = \frac{2\pi(r-m)}{m\Delta x} \qquad\qquad \frac{m}{2} \le r < m$$

29

$$l = \frac{2\pi t}{n\Delta y} \qquad 0 \le t < \frac{n}{2} \qquad (7)$$

$$l = \frac{2\pi(t-n)}{n\Delta y} \qquad \frac{n}{2} \le t < n$$

then the FFT routine will return the Fourier transform as de-
fined in Eq. (2). The wavenumbers associated with the trans-
form are given by Eqs. (6) and (7).

The DMA data tapes, however, do not contain values
associated with boundary points of the area in question. In-
stead, the data are associated with averages over elementary
rectangles and lie in the interior of the area as illustrated
by Figure 7. It is also necessary that the number of data
points along each spatial dimension be an integer power of 2.
In order to take account of this displacement of the data from
the boundary points, a redefinition of the Cooley-Tukey coeffi-
cients is required:

$$a_{su} = \frac{m\Delta x n\Delta y}{4\pi^2} \, h[(s+\tfrac{1}{2})\Delta x, \ (u+\tfrac{1}{2})\Delta y] \qquad (8)$$

Following the calculation of the FFT coefficients a subsequent
calculation to form the topography transform is carried out:

$$\hat{h}_{kl} = \tilde{a}_{rt} \, e^{-\tfrac{1}{2}i(k\Delta x + l\Delta y)} \qquad (9)$$

This transformation of terms allows one to approximate the
integral of Eq. (2) without having to use boundary points.

The steps of the spectrum function calculation, conse-
quently, are as follows: The topography data are interpolated
to obtain new values corresponding to the next higher integer
power of 2 equally spaced points in each direction. These .

values are pre-processed according to Eq. (8). The FFT is executed and the resulting array is then processed as in Eq. (9) to obtain the Fourier transform of the height. The spectrum function is then straightforwardly calculated according to Eq. (1).

The spectrum function, together with the associated k,l wavenumbers, is written into a file for future use. The final task of Module 1 is to print a tabulation of $A(k,l)$. Contour maps of the topography and spectrum function are also plotted. Additionally, a contour map of the log of the spectrum function is made.

4.4   SPECTRUM FUNCTION CALCULATIONS

Since the spectrum function code was completed late in the last contract period, only a few example calculations of the topography function A have been completed. The study consisted of an examination of ten geographic areas in the United States differing widely in topography and location. Topography data from the DMA magnetic tapes having 5' resolution were used and the resulting spectrum functions were calculated. These same topographic data were then resolved to 10', by combining 5' values, and the spectrum functions were determined again. The resulting set of data can be investigated to determine the behavior of A as a function of differing topographies and also determine the sensitivity of A to topographical resolution.

A representative sample of terrains was selected; these data consisted of contiguous rectangles within a strip of land across the U.S. from 34°20' to 37°0' north latitude and from 94°20' to 121° west longitude. This strip includes the rugged topography of the High Sierra and Rocky Mountains, the flat plains of the midwest, and the Appalachian mountains of the east.

The strip was composed of 10 contiguous topographical grids 2°40' on a side (corresponding to 32×32 data points for each grid at 5' resolution). This set of grids constitutes a preliminary cross-section of characteristic topographies.

Since the study is still in progress, only a brief examination of a sample grid is currently available. Figures 8-11 show contour maps of one geographic area and its associated spectrum function located in the rectangle 118°20' to 121° west longitude, 34°20' to 37°0' latitude at both 5' and 10' resolution. This area corresponds to the first grid in the set of grids and encompasses portions of the San Joaquin Valley, Sierra-Nevada mountains and the coastal ranges of California. Figure 8 presents the contour map at 5' resolution and Figure 9 is the corresponding calculated spectrum function.. Figure 10 presents the contour map for the 10' resolution data. Figure 11 is the corresponding spectrum function.

The figures contain computer produced contours similar to those produced by the HAIFA code in previous reports.[4] Actual contours could be constructed by connecting like numbers; the actual value (h or A) corresponding to the plotted numbers are given by the formula

$$V_{actual} = V_{min} + (V_{max}-V_{min})[(N+\tfrac{1}{4})\pm\tfrac{1}{4})]/10 \quad.$$

The quantities V_{max}, V_{min} refer to maximum and minimum values of the variable and N is the plotted symbol. The $\pm\tfrac{1}{4}$ indicates the range in V_{actual} corresponding to a particular symbol. The blank fields appearing in the plots correspond to contour levels intermediate between the symbols which are left blank for visual clarity. A(0,0) appearing on the spectrum

H_{max} = 3.559 km

H_{min} = 0.000 km
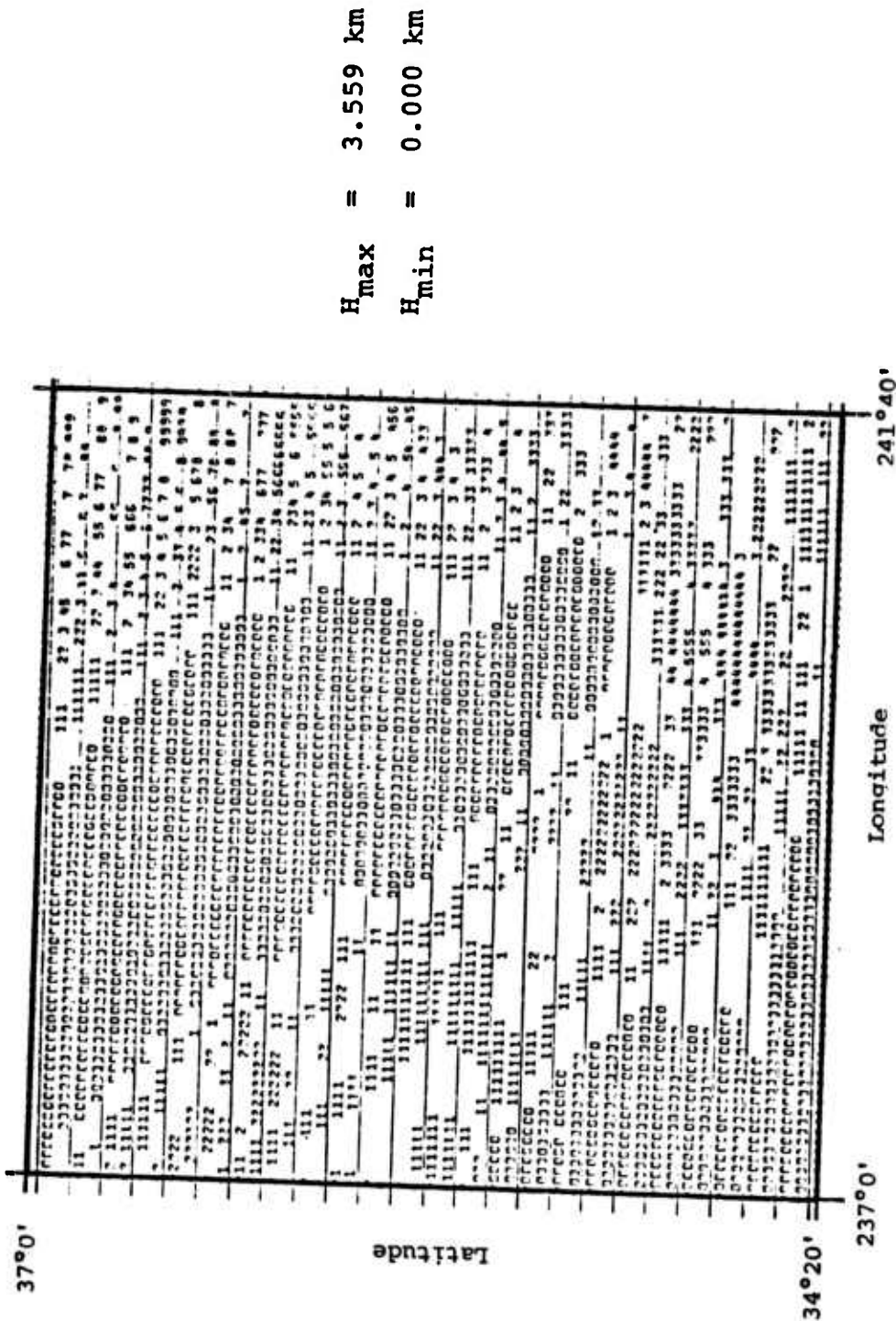
Figure 8   San Joaquin-Sierra Nevada topography contour map derived from 5' resolution data.

33

$A_{max} = 1.07 \times 10^2 \ km^3$

$A_{min} = 1.36 \times 10^{-5} \ km^3$

$A(0,0) = 6.69 \times 10^2 \ km^3$
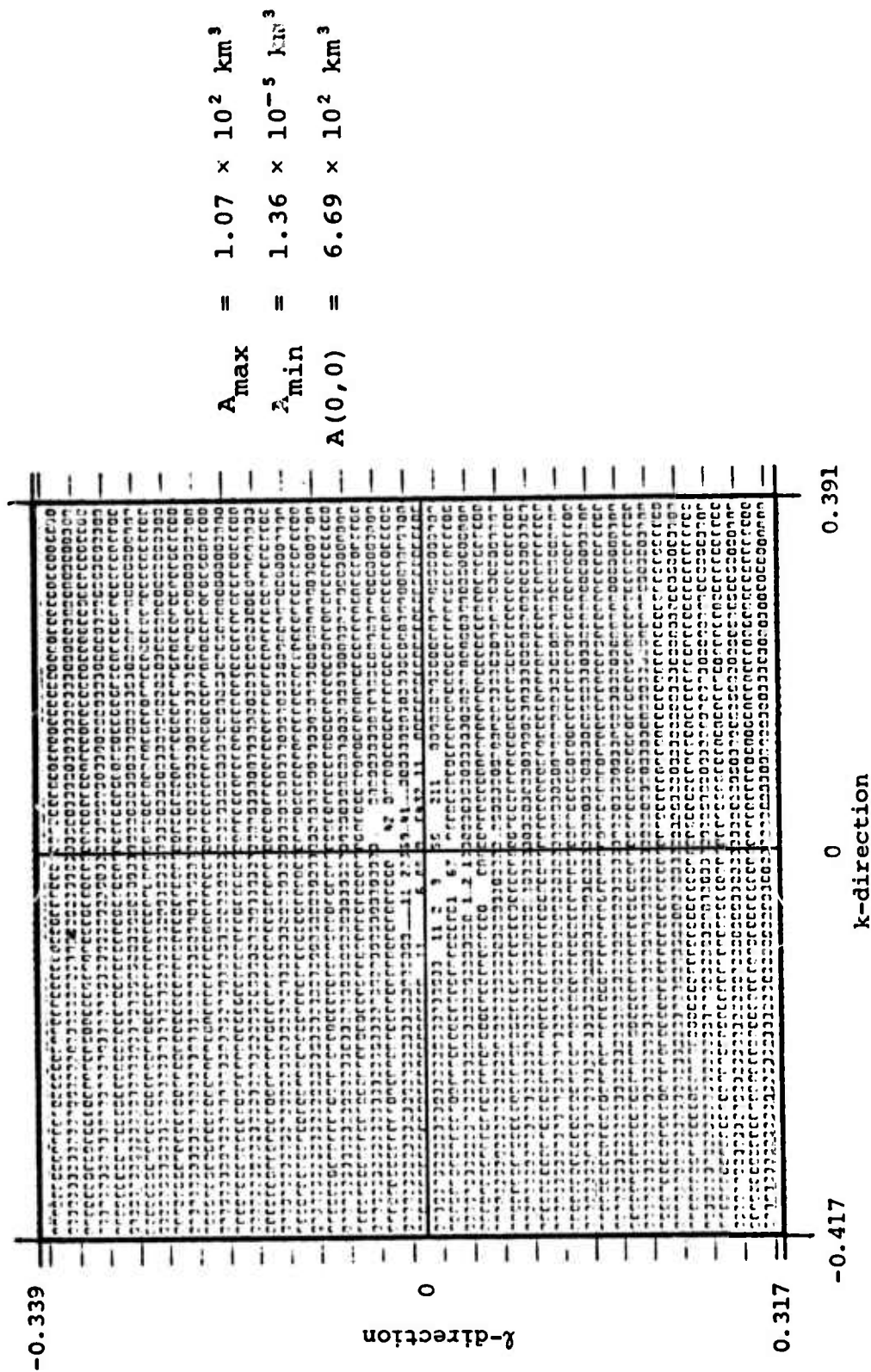
Figure 9   Spectrum function distribution using topography data from Figure 8.

34

$$H_{max} = 3.428 \text{ km}$$

$$H_{min} = 0.000 \text{ km}$$
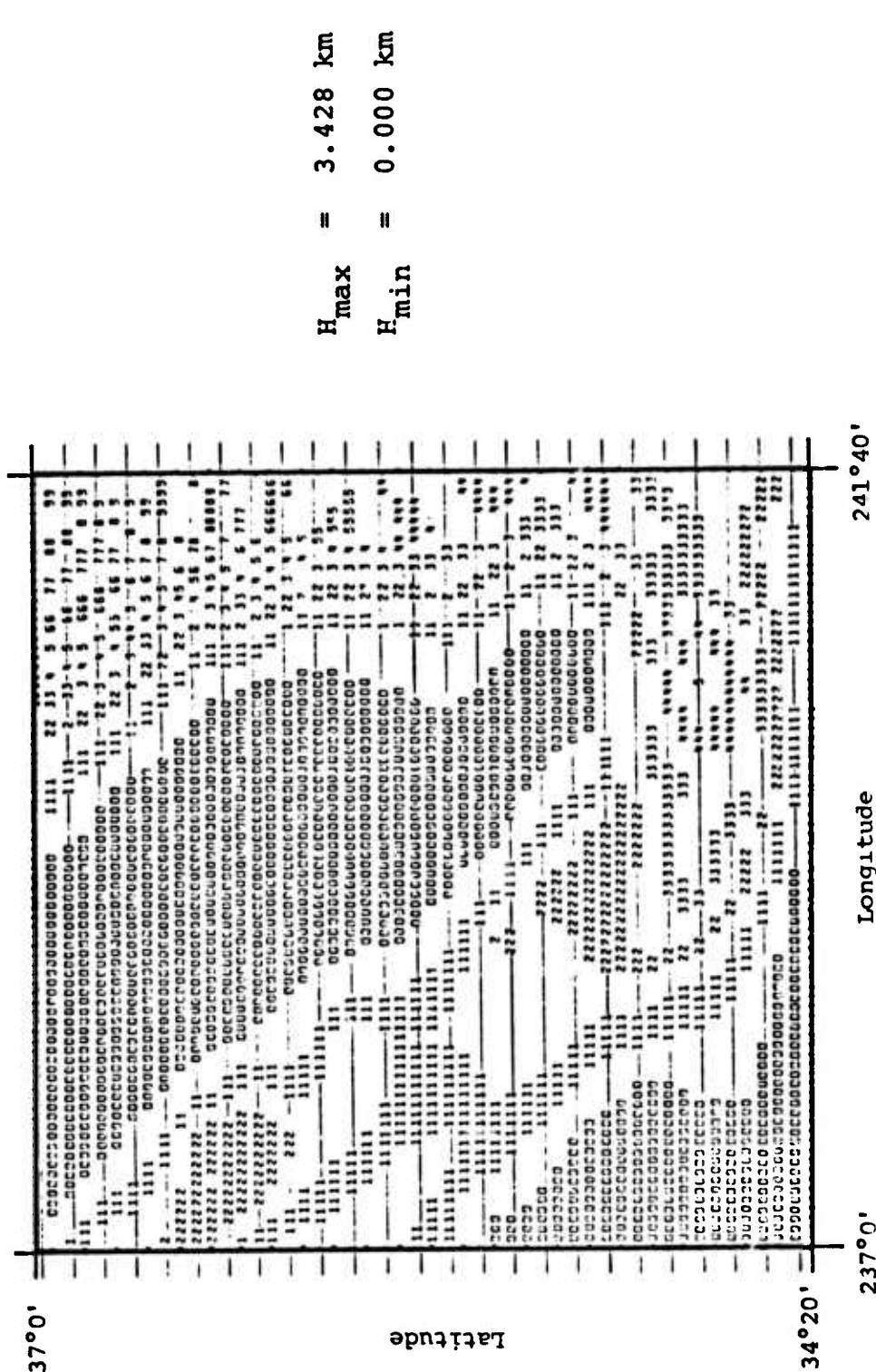
Figure 10 San Joaquin-Sierra Nevada topography contour map derived from 10' resolution data.

35

$A_{max} = 1.08 \times 10^2 \ km^3$

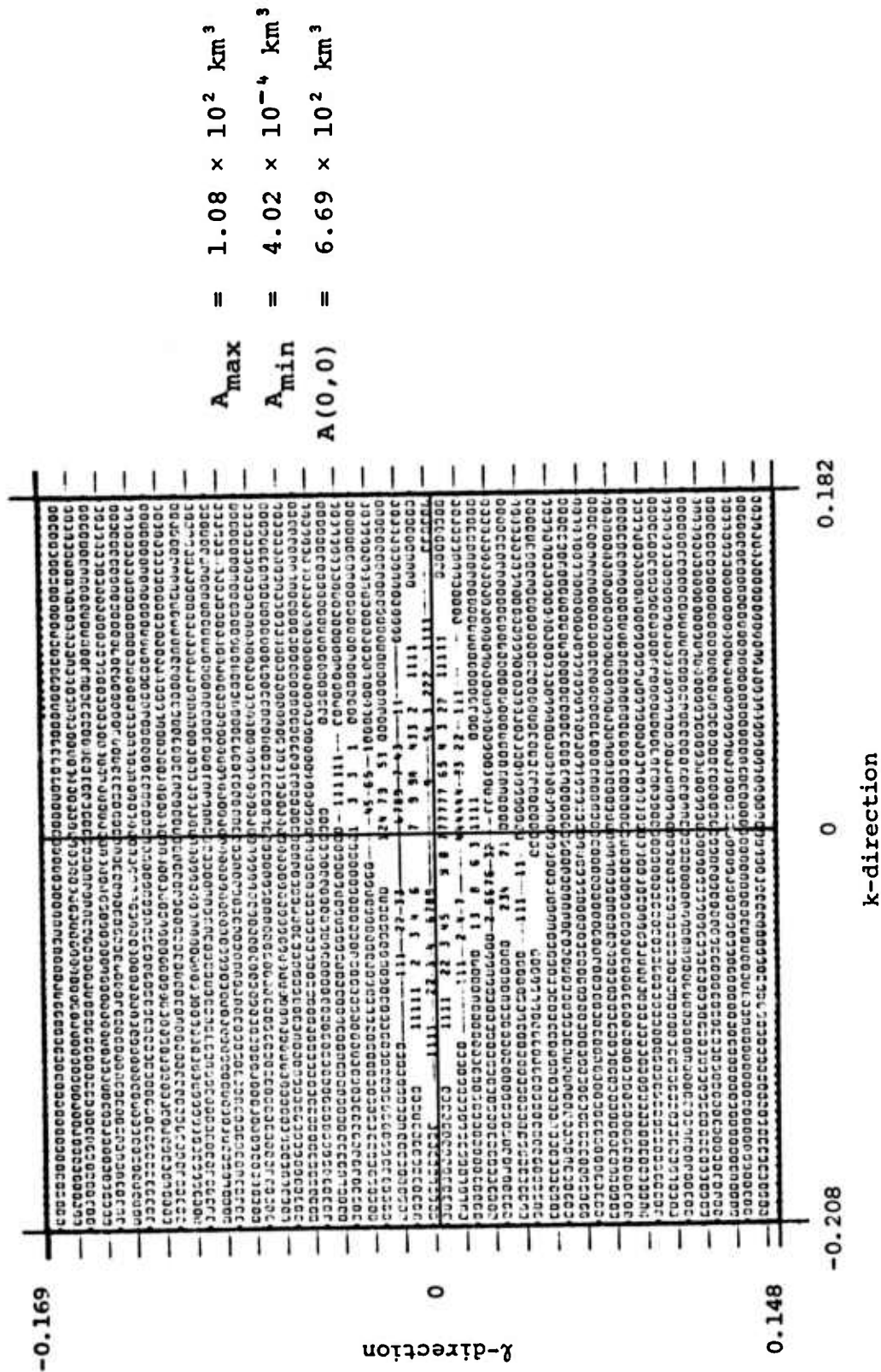$A_{min} = 4.02 \times 10^{-4} \ km^3$

$A(0,0) = 6.69 \times 10^2 \ km^3$

Figure 11   Spectrum function distribution using topography data from Figure 10.

function plots is the value of the spectrum function at k=0, l=0 . Due to its large value it has omitted in the contouring so that resolution could be improved.

The presence of the San Joaquin Valley (having a pronounced north-west to south-east axis) in the grid introduced an interesting asymmetry to the topography. Upon examination of the graph we find that these regularities in the topography are reflected in the spectrum function. The topography consists primarily of two ridges rising above a flat plain. One feature extends north-south near the right edge of the topography graph and the other crosses the graph at approximately a 45° angle. These two ridges correspond to the two finger-like extensions seen in the spectrum function plot of Figures 8 and 10. The horizontal extension corresponds mainly to the irregularities in the N-S ridge while the diagonal arm reflects the diagonal ridge.

Due to the symmetry properties of A, the graphs exhibit a reflected symmetry through the origin. (Note: Due to the number of contour intervals and the slight displacement of the graphs of topography and spectrum function, exactly the same plot character may not appear in the reflected plot.) We also see the same characteristic features appearing in both A plots. The resolution of Figure 9 hampers further comparisons between the spectrum functions corresponding to 5' and 10' resolution. Work is under way to improve this.

Examination of the other topographic regions of the set reveals that frequently a prominent topographical feature will be reflected in a clearly observable corresponding feature of the A distribution. Further investigation of these data is currently in progress.

4.5   COMPUTER CODE FOR REYNOLD'S STRESS CALCULATION

4.5.1   The Equations

The equations incorporated in the code are substantially those of Bretherton.[1]   A slight modification has been made to the trapped wave integrals, however. A factor of κ in the numerator of the sums which was omitted by Bretherton, has been incorporated into the code.

4.5.2   Numerical Method

The numerical methods discussed in the previous report[2] have undergone considerable modification. A faster, more accurate method of locating trapped waves was developed and incorporated. It was felt that in some atmospheric situations the code, in its present configuration, might miss a wave. Another consideration was the optimization of the code such that a large scale survey study could be carried out more economically. Calculations to date indicate that a reduction of the cost of a calculation by a factor of two or more has been achieved as a result of these steps. Even greater savings are expected once the code modification program is completed.

In addition, the calculation is considerably more self-contained and accurate. The logic of the new version of the code is outlined in the flow chart of Figure 12. Since the code is radically different in structure from the previous version, a detailed description is also presented in the following section.

A trapezoidal integration scheme is currently being employed in the evaluation of the stress integrals. The integration uses a constant increment $\Delta\kappa$ and $\Delta\phi$ of κ and ϕ . The integration over κ is currently carried out for each value of ϕ , although the order of calculations will be changed for
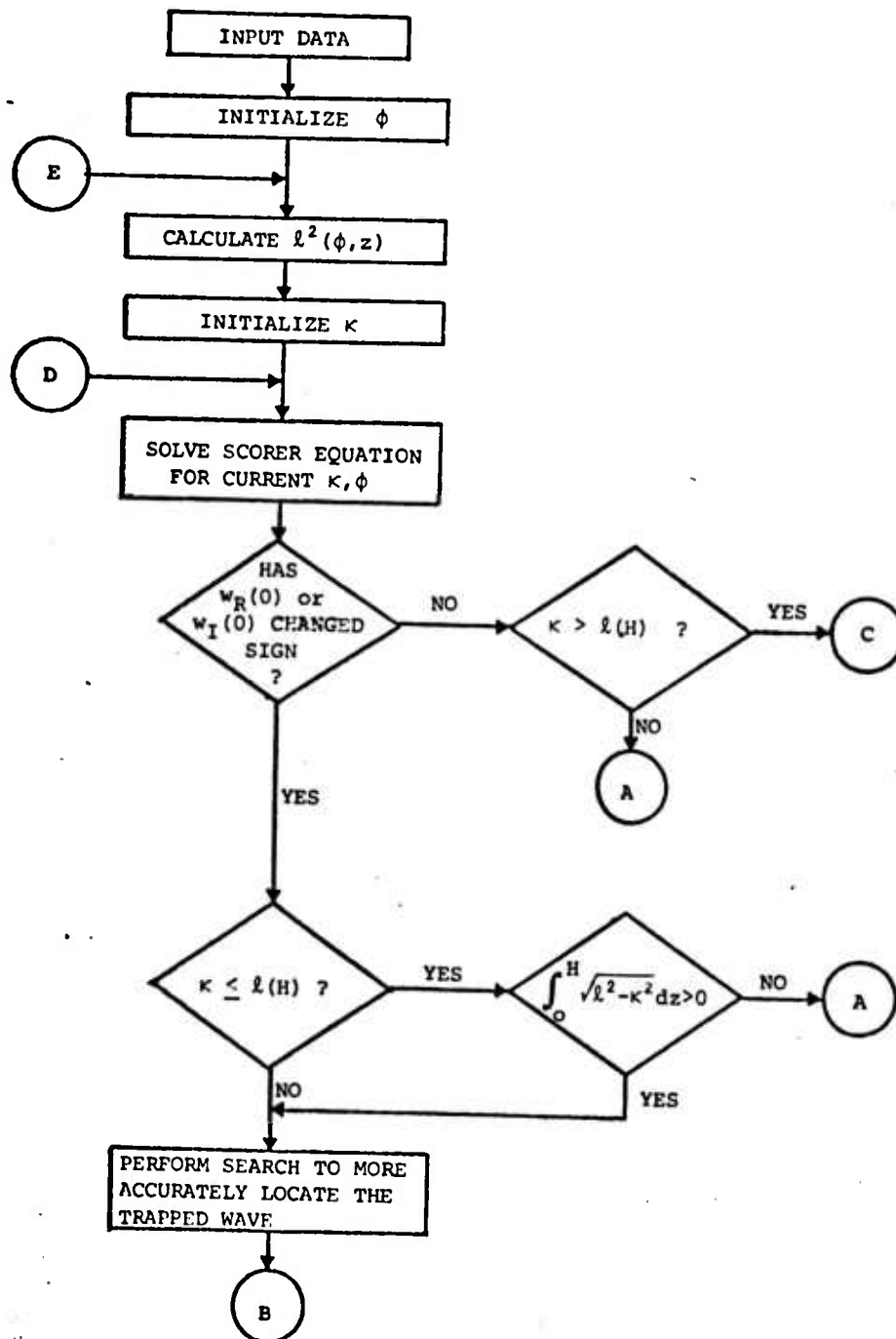
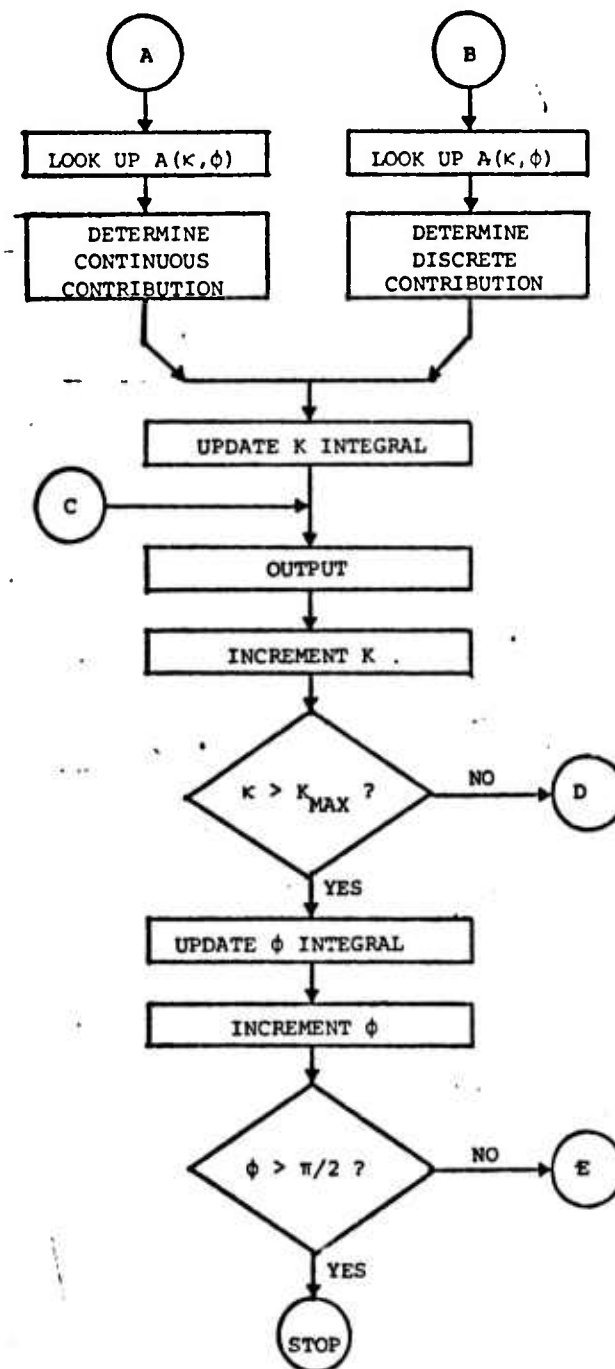Figure 12   General flow diagram of drag code.

39

Figure 12   contd.

40

the newly organized code. First, the Scorer parameter profile is calculated for the current value of ϕ. Values of $\hat{W}$ are then successively calculated for all κ-values. The values at $z=0$ are then tested to determine whether there are nearby trapped waves. If the test is negative and $\kappa \leq \ell(H)$ then the region is defined to be part of the continuous spectrum and the stress integrals are incremented using Bretherton's Eq. (54). If the criterion for a trapped wave is met, a search to localize it more accurately is begun. κ-values between $\kappa - \Delta\kappa$ and κ are searched until the denominator of the F expression is minimized to within a user-prescribed accuracy. If $\kappa > \ell(H)$, the trapped wave occurs at a κ-value for which the perturbation velocity $\hat{w}$ at $z=0$ vanishes. Since the real and imaginary parts of the solution for $\kappa > \ell(H)$ are proportional to each other, they both must vanish together. This property simplifies the search scheme for the trapped wave considerably.

Figure 13 depicts the typical κ-dependence of the Scorer equation solutions at $z=0$. Let κ_i be the κ-value of the trapped wave. We also denote by w_1 and w_2 the real parts of the Scorer equation solution for $\kappa - \Delta\kappa$ and κ at $z=0$. An iterative search is performed which utilizes linear interpolation between w_1 and w_2 to obtain a more accurate guess at κ_i. Using this value of κ and its corresponding $w_R(0)$ another linear interpolation is performed using the member of the old pair of w's of opposite sign to the new value. The search converges quite rapidly and usually requires only a few iterations. The criterion for convergence of the search is a user-specified value of the permissible change in the "s" term of Bretherton's Eq. (61). It was found to be superior to a test of $w_R(\kappa,\phi)$ compared with a small number which was found to be unreliable due to possible rounding errors
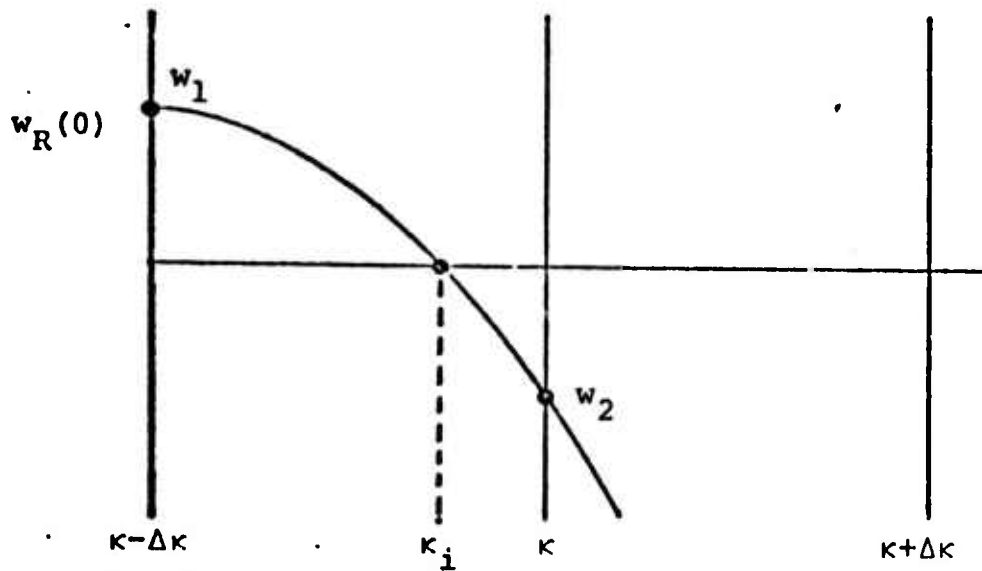
Figure 13   Typical dependence on wavenumber of the solution
of the Scorer equation at z=0, $w_R(0)$ vs. κ.

in $\hat{w}$ where the amplitude of $\hat{w}$ is very large. Care must be taken in choosing the size of $\Delta\kappa$ since it might be possible for $\hat{w}(\kappa,\phi)$ to change sign twice within the interval and a trapped wave would be missed. Present $\Delta\kappa$ are very conservative, and it is felt that this situation will not arise very often.

For values of $\kappa \leq \ell(H)$, the real and imaginary parts of the solution to Scorer's equation are linearly independent. Consequently, the real and imaginary parts will not simultaneously vanish at $z=0$ for any κ-value. Both may simultaneously become quite small, however. When this condition occurs, a peak in the F distribution will be present. The height of the peak is a function of $\hat{w}^*(0)\,\hat{w}(0)$, and its "Q" , corresponding to the width of the peak, is a function of the depth of the region in which the solution of the Scorer equation undergoes exponential decrease. A large value of "Q" corresponds to trapped waves with a very slight upwards energy leak.

Consequently, a trapped wave in the region of $\kappa \leq \ell(H)$ must satisfy two conditions. First, the value of $\hat{w}^*(0)\,w(0)$ must have a local minimum. And second, the peak in the value of F must be sharply localized. Bretherton suggests that the first condition can be recognized by examining the quantity $\arg[\hat{w}(\kappa,\phi)]$ for sudden changes as a function of κ . This, in effect, requires either the real or the imaginary part of $\hat{w}$ at $z=0$ to change sign. This is a necessary condition for a trapped wave to exist. For $\kappa > \ell(H)$ it is also a sufficient condition. However, for $\kappa \leq \ell(H)$ the second condition discussed above must also be met. A sharply localized peak of F requires that a resonant behavior of $\hat{w}$ occur. This, in turn, calles for a sufficiently deep atmospheric layer in which gravity waves are reflected so that the leakage of energy into the stratosphere is small. This condition can be quantitied in

43

terms of the quantity

$$\int_0^H \sqrt{\ell^2(z) - \kappa^2}\; dz \quad,$$

which is a measure of the exponential decrease of $\hat{w}$ due to wave reflection. When

$$\int_0^H \sqrt{\ell^2(z) - \kappa^2}\; dz > \sigma$$

where σ is a user-specified number, the resonance in F is judged to be sufficiently sharp to perform the integration over it analytically. If the above condition is not met, the integration is to be performed numerically. In this case, the integrand changes sufficiently slowly that an accurate numerical quadrature can be performed with a reasonable integration step $\Delta\kappa$.

Since the extent and strength of the region of exponential decay in the solution to Scorer's equation increases with increasing κ , the resonances (if any) lying below $\kappa = \ell(H)$ will exhibit sharper and sharper peaks as κ increases. Through the choice of σ the user has control over the extent of the spectrum to be considered as continuum. A small value of σ will require that rather broad lines are treated analytically, while a large value will treat quite sharp lines as continuous.

Tests of this section of the code will be carried out for several choices of atmospheric parameters in order to determine an optimum value of σ for accuracy and speed of calculation.

5. FUTURE PLANS

Now that the computer codes for evaluating the linear steady-state Reynold's stress are essentially complete, attention will be turned to the calculation and parameterization phases. A full scale survey study will involve several hundred topographical areas and several dozen typical atmospheric conditions. Since each combination of an atmosphere with a topographic area constitutes a particular case, it is seen that the total number of cases under consideration is the product of the number of atmospheres and the number of topographies. This number may exceed several thousand.

In order to perform these calculations in an efficient manner, we have examined the time requirements of the codes and have considered the organization of the calculational sequence. The linear form of the equations is the key to the most efficient organization. In order to avoid duplicate calculation of topography factors and atmospheric factors, it is desirable to modularize the steady-state code. Rather than calculating the topographic factor, atmospheric factor, and Reynold's stress integral on a case-by-case basis, it is desirable to form and tabulate the topographic and atmospheric factors separately. The desired stress integrals can then be found by combining the factors in an inexpensive numerical integration. This method results in major savings in computer cost.

The spectrum function for a typical 32×32 point topography requires a calculation on the UNIVAC 1108 of 10 seconds CPU time. The calculation of a typical atmospheric factor F requires 1100 seconds on the same computer. For the survey study of approximately 100 topographies and 10 atmospheres, the case-by-case calculation would require computer CPU time of approximately 10^6 seconds. A modular approach would require approximately 10^4 seconds for the atmospheres and 10^3 seconds for the topographies. Preliminary estimates are that the Reynold's stress integrations will also require approximately 10^4 seconds. The total time would then be $\sim 2 \times 10^4$ seconds, of which only a minor part is required for the topography Fourier transforms.

The modulization revisions are currently being made and will be completed shortly. In more detail, the linear steady-state calculation is divided into three segments. Communication between the segments is implemented through the use of information files on "fast" drums. The first segment has been completed and calculates and stores the spectrum function distribution for a given grid. This code is described more fully in Section 4. The second segment calculates the topographic factor, F, for a complete range of values of κ, ϕ. The resulting distribution is stored on mass storage along with the corresponding discrete coordinate values. This code is essentially complete and is substantially the same as the Reynold's stress code described in Section 4. Finally, the third segment utilizes the results of the first two to perform the integration of the stress integrals. Since all of the data are available simultaneously, a higher order quadrature scheme can be utilized.

This organization of the calculation offers several advantages. Only specifically desired cases need be integrated

and the files can be augmented with additional cases as desired. The individual segments of the code can also be executed for a lower computing rate since a much smaller portion of core is required than for the code version in which all parts are unified. The computer instructions for each segment require about 10,000 words of core storage. The storage file for the spectrum function requires several thousand words per grid. The F factor file requires around 10,000 words per atmosphere. From these numbers it can be estimated that the mass storage requirements will not be expensive. Computation times will be slightly longer due to accessing the drums, but overall rates will be cheaper on a case-by-case comparison. Greater flexibility is also achieved since additional atmospheres or topographies can be easily added.

REFERENCES

1. Bretherton, F.P., "Momentum Transport by Gravity Waves," Qtrly. J. R. Met. Soc. (1969), 95, p.213.

2. "The Effects of Meso-Scale and Small-Scale Interactions on Global Climate," Report No. SSS-R-73-1727, Contract No. DAHC04-73-C-0003 (15 June 1973), Systems, Science and Software, La Jolla, California.

3. Sheridan, R.S., W.G. England, B.E. Freeman, and J.R. Taft, Systems, Science and Software, "A Computerized Simulation of Flow over the Sierra Nevadas," Summer Simulation Conference, Montreal, Quebec, Canada, 17-19 July 1973.

4. "The Effects of Meso-Scale and Small-Scale Interactions on Global Climate," Report No. 3SR-1034, Contract No. DAHC04-71-C-0018 (31 March 1972), Systems, Science and Software, La Jolla, California.

SYSTEMS, SCIENCE AND SOFTWARE

PART II

DEVELOPMENT OF TRANSIENT MODELING

49

1. INTRODUCTION

The previous project report[1] presented the derivation of the equations solved in the newly-developed 3-D Boussinesq approximation fluid dynamics computer code STUFF. Modeling capabilities were discussed and to some extent the numerical approach taken in STUFF was examined. Since then, the code has undergone considerable modification. Efforts have been primarily directed toward code optimization in execution in order to increase computation speed. However, some attention was given to reducing the core requirements of the instructions.

The following discussion records the results of this modification program. Also, a more detailed account of the numerical techniques now employed in STUFF is given. This latest version of the computer code is designated as STUFF3.

2. MODIFICATIONS TO THE STUFF3 CODE

2.1 CODE OPTIMIZATION

2.1.1 Introduction

At the close of the last contract period, it was determined that several parts of the STUFF3 code might be significantly speeded up by more efficient programming. Timing runs had indicated that the major contributors to total execution time were the local macro-scale calculation and the particle buffering scheme. As a result, the routines which transfer the particles to and from core were completely rewritten. The macro-scale calculation was more efficiently coded, and a scheme to selectively avoid the full macro-scale integration for some time steps was introduced. The results of this program are presented below.

2.1.2 Modifications — The Buffering Scheme

To review, STUFF3 utilizes a particle buffering scheme such that only a small fraction of the Lagrangian particle properties occupy core storage at any one time. The rest reside on mass storage devices in distinct batches. They are retrieved, when required, recalculated, and subsequently returned, to mass storage. This method allows a certain degree of machine independence, since the core requirements of the code are reduced drastically. Currently, STUFF3 has been

utilizing a particle density of eight particles per Eulerian cell. This results in a requirement of 64,000 particles in a 20×20×20 cell grid. Since each particle requires six storage locations for position, density, contaminant and turbulent energy concentration, one sees that this amounts to 384,000 required storage locations. To keep such a large amount of information in core would be prohibitively expensive.

With the advent of mass storage devices with high rates of information transfer, buffering schemes have become practical, both from an economical and run time standpoint. STUFF3 stores the particle variables, associated with 500 particles, in core at one time. As a result only 3000 core locations are required to handle the particle-based information. Since most accounting systems utilized by computer installations heavily weight core utilization, this technique results in marked economies. The penalty one pays is in the resultant increased run times and a high use of the I/O channels to the peripheral devices. Both of these factors increase the cost of a calculation.

Timing runs were made which indicated that the buffering scheme was, indeed, adding substantially to run costs. As a result, an examination of the code was made to determine if the particles contained in core were being utilized to their maximum efficiency. By alteration of the order of the particle calculation routines, it was found that the number of exchanges of particles between core and mass storage per time step could be reduced from 9.5 to 3.5. The effect of this reduction has been to render run costs considerably less dependent on the buffering feature; the contribution currently is approximately 15 percent. The reason for this small value is that arithmetic calculations now dominate the time required to execute a time step. Actual run time and cost have been reduced by more than

30 percent on an average basis. Larger grids yield larger relative savings.

2.1.3   Modifications — The Macro-Scale Calculation

Further optimization was centered around the calculation of the local turbulent macro-scale length. Since this requires three-dimensional integrations over all space and calculation of the components of the strain-rate tensor for every cell, a large fraction of the total run time can be attributed to it. Two optimizing mechanisms were employed. The first was a rather straightforward implementation of a suggestion presented in the last report,[1] namely, that the characteristics of the local mean flow are influenced primarily by the quantities in the neighborhood of the point in question, and less by quantities further away. This scheme was incorporated into the optimization program as a limitation on the number of surrounding cells to be included in the spatial integration. This limit is established by the user, as an input number.

The second scheme is suggested by the fact that the code is used to study the evolution of fluid flow from a non-steady to steady-state configuration. As the fluid progresses through the evolutionary sequence, one would expect that the local macro-scale also reach a steady-state since they are, in effect, dependent on the mean flow. Also, we would not expect large changes in the eddy viscosity distribution unless the macro-scales have changed significantly. The user specifies a value which he considers to represent a significant change from one time step to the next in the macro-scale distribution. If this value is exceeded on any time step, then a complete re-integration occurs. If not, then the eddy viscosities are boosted at a rate which corresponds to the rate of change of the average macro-scale. This procedure has the

advantage of entirely eliminating the integration at late times resulting in correspondingly more rapid executions of each time step.

2.1.4   Modifications — Reduced Instructions

Secondary optimization focused on the reduction of the number of machine language instructions necessary to represent the FORTRAN coding. To effect this, all routines were examined for FORTRAN structures which could be re-coded such that they would be translated by the compiler more efficiently. This re-coding would lead to an overall reduction in the number of core locations necessary for code instructions. Upon completion of this task, the code was reduced by nearly 5000 words to 18,000 words, a reduction of over 20 percent.

2.2   NUMERICAL METHOD

2.2.1   The Equations

Before describing the numerical techniques employed in STUFF3, it is useful to summarize the set of equations presented in the previous report. These are the fluid dynamic and turbulence equations governing the flow. They are, however, in a somewhat awkward form for numerical solution. A secondary set is, therefore, presented which corresponds to the approach incorporated into STUFF. The generalized set is:

$$\frac{\partial}{\partial x_i} (U_i) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(U_i) + \frac{\partial}{\partial x_j}(U_i U_j) = \frac{\partial}{\partial x_j}\left[(\nu + \alpha\lambda\sqrt{2E})\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right]$$

$$- \frac{\partial P}{\partial x_i} + Q g_i + s \tag{2}$$

$$\frac{\partial}{\partial t}(E) + \frac{\partial}{\partial x_j}(U_j E) = \alpha\lambda\sqrt{2E}\,\Omega^2 - \beta(2E)^{7/6}\,J^{1/3}$$

$$+ \frac{\partial}{\partial x_j}\left(\alpha\gamma\lambda\sqrt{2E}\,\frac{\partial E}{\partial x_j}\right) - \alpha\gamma\lambda\sqrt{2E}\,g_j\,\frac{\partial Q}{\partial x_j} \tag{3}$$

$$\frac{\partial}{\partial t}(Q) + \frac{\partial}{\partial x_j}(U_j Q) = \frac{\partial}{\partial x_j}\left[(\alpha\lambda\gamma\sqrt{2E}+D)\,\frac{\partial Q}{\partial x_j}\right] + \pi_Q \tag{4}$$

Equations (1-3) represent conservation of mass, mean flow momentum, and turbulent kinetic energy, respectively. The fourth equation represents the thermal energy transport equation written in terms of a density variation. Additionally, with a redefinition of Q and D, Eq. (4) provides for the tracing of an arbitrary passive contaminant. All terms have been defined in the previous report. The heuristic coefficients α, β, and γ are given by

$$\alpha = 0.065\left\{1 + \exp\left[-\left(\frac{y}{\lambda} - 1\right)^2\right]\right\} \tag{5}$$

$$\frac{1}{\beta} = 3.7\left\{1 + \exp\left[-\left(\frac{y}{\lambda} - 1\right)^2\right]\right\} \tag{6}$$

$$\gamma = 1.4 - 0.4\,\exp\left[-\left(\frac{y}{\lambda} - 1\right)^2\right] \tag{7}$$

The local turbulent macro-scale is given by

$$\lambda^2(\vec{x}) = I^2(\vec{x})/J^2(\vec{x}) \tag{8}$$

where

$$I^2(\vec{x}) = \int_{\substack{All \\ Space}} w(\vec{x},\vec{x}')\Omega^4(\vec{x}')\,dv' \qquad (9)$$

$$J^2(\vec{x}) = \int_{\substack{All \\ Space}} w(\vec{x},\vec{x}')\left[\Omega\Omega'(\vec{x}')\right]^2\,dv' \qquad (10)$$

and $w(x,x')$ is a normalized Gaussian weighting function defined as

$$w(\vec{x},\vec{x}') = \frac{\exp\left[-\dfrac{(\vec{x}-\vec{x}')\cdot(\vec{x}-\vec{x}')}{\lambda^2(\vec{x})}\right]}{\displaystyle\int_{\substack{All \\ Space}} \exp\left[-\dfrac{(\vec{x}-\vec{x}')\cdot(\vec{x}-\vec{x}')}{\lambda^2(\vec{x})}\right]\,dv'} \qquad . \qquad (11)$$

Additionally,

$$\Omega^2 = \frac{1}{2}\,\Gamma_{ij}\Gamma_{ij} \qquad (12)$$

$$(\Omega')^2 = (\partial\Omega/\partial x_i)(\partial\Omega/\partial x_i) \qquad . \qquad (13)$$

While Eq. (2) does, in fact, govern the evolution of momentum distribution in the system, it contains a pressure gradient term which is awkward to determine. As a result, many fluid dynamic computer codes use a formulation involving derived variables which eliminate the direct pressure calculation. Methods using the stream function, for example, require special care in applying boundary conditions, particularly at internal

boundaries. A method of Chorin[2] is free of these difficulties while still avoiding the calculation of the pressure directly. The resultant set is expressed in terms of primitive variables, and, hence, allows straightforward application of boundary conditions. Equations (1) and (2) are combined, and use is made of the assumption that a pressure field alone will not generate rotational flow. The resulting momentum and mass conservation equations appear as:

$$\frac{\partial}{\partial t}(\tilde{U}_i) + \frac{\partial}{\partial x_j}(\tilde{U}_i U_j) = \frac{\partial}{\partial x_j}\left\{(\nu + \alpha\lambda\sqrt{2E})\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right\}$$

$$+ Qg_i + S \tag{14}$$

$$\frac{\partial^2}{\partial x_i^2}(\psi) = -\frac{\partial}{\partial x_i}(\tilde{U}_i) \tag{15}$$

$$U_i = \tilde{U}_i + \frac{\partial}{\partial x_i}(\psi) \quad . \tag{16}$$

With this modification, the equations are in the final form for computation. It now remains to discuss the numerical method.

2.2.2   Numerical Method — Overview

STUFF3 is a mixed-mode hydrodynamics code in which both an Eulerian and Lagrangian matrix representation exist concurrently. In general, mixed-mode codes attempt to minimize the inadequacies of one representation by combining the best features of both. This usually results in some or all of the field variables being carried in both representations (i.e., available in either particle or cell based quantities), with

one or the other being utilized as the solution progresses, to evaluate the various terms appearing in the differential equations. At some point there must exist an exchange of data between the two such that each may be able to exert a "corrective" function upon the other. It is through this mechanism that distortions in the Lagrangian net, such as those due to high amplitude waves, can be minimized and that artificial diffusion in the Eulerian sense is greatly reduced.

Before examining the data exchange routines of the STUFF code, a discussion of the allocation of the dynamic variables between the two representations is in order.

All of the scalar fields (density, contaminant concentration, and turbulent energy) are assigned to both representations, while the vector fields (in our case this is just the velocity field) are associated with the Eulerian grid alone. All changes in the distribution of a scalar dynamic variable are reflected in changes in the particle based values during a time step. At the completion of the evaluation of all differential terms, the Eulerian quantities are assigned the "average" of the values associated with particles contained within the cell. This censusing of the particles is one of two exchanges of data in the STUFF code. The purpose is primarily to replace the explicit Eulerian evaluation of the scalar advection terms with a more accurate Lagrangian one. In this manner, the errors induced by artificial diffusion are avoided. This censusing procedure is defined more explicitly in Section 2.2.3.

The second exchange of data takes place in the calculation of the scalar diffusion terms. These terms use both particle and cell based values. This procedure is also defined more explicitly in Section 2.2.3.

2.2.3 Numerical Method — A Closer Look

Perhaps the best method to describe the numerical techniques of STUFF3 is to detail the steps of a typical cycle of calculation.

Leaving the question of initialization to Section 2.2.4, we assume that sufficient data are available to begin a time step. The calculational sequence is as follows:

STEP 1 Calculate the local eddy viscosity.

STEP 2 Evaluate source and diffusion terms in the scalar equations.

STEP 3 Partially update particle-based scalar variables by adding source and diffusion terms.

STEP 4 Solve the momentum equations utilizing method of Chorin.

STEP 5 Move the particles with the new velocity field and evaluate the scalar advection terms.

STEP 6 Remove or add particles as needed to account for sources or sinks of momentum.

STEP 7 Establish new cell-based values from a particle census.

STEP 8 Advance the time.

STEP 9 Honor output requests.

STEP 1 — Calculation of the local eddy viscosity.

The local eddy viscosity is calculated in routine EVMAKE. Additionally, the turbulent energy source terms involving Ω and J are calculated and stored in temporary storage. The diffusion-limited time step is also calculated. Finally, a "boosted" eddy viscosity is constructed for use in the momentum equations. This quantity is the larger of the

true eddy viscosity and a "stability" viscosity. The latter is the smallest viscosity needed to stabilize the weak instability arising from the finite difference operator used in the momentum equation. The sole purpose of this procedure is to insure computational stability.

The calculation of the eddy viscosity requires a knowledge of the local macro-scale distribution. Hence, EVMAKE calculates this as well. At this point a feature of the previously described code optimization is introduced. A value is assigned by the user which is a measure of the error that will be tolerated in the eddy viscosity calculation. The average macro-scale of the grid is calculated on each cycle. In order for the local macro-scale to be spatially integrated, the current average macro-scale must differ from the previous one by the given significance level. If the test fails, then the old local eddy viscosities are boosted by an amount corresponding to the rate of change in the average macro-scale. This procedure is found to give excellent results in practice, showing good agreement with the continuous integration procedure. The largest deviation appears in regions of abrupt changes in the boundary, as would be expected.

The user may also specify the maximum extent of the spatial integration of I^2 and J^2. This allows substantial reduction in computing time with only a moderate loss in accuracy since the integrands fall off strongly at larger distances.

## STEP 2 — Evaluation of scalar source and diffusion terms.

In this calculation, the cell-based scalar field values and eddy viscosity distribution are utilized to construct the diffusion term contributions to the scalar equations. To these terms are added any scalar source terms arising from sources or

sinks. In the case of turbulent kinetic energy, the sources calculated in routine EVMAKE are included. The result is to create an effective "source" rate for each of the cells. This is stored temporarily for use in the next step.

STEP 3 — Partial update of particle scalar quantities.

Some difficulty is encountered in using the diffusion and source rates to update the particle scalar quantities for a time Δt. The diffusion and source rates are cell-centered quantities. A first approximation would be to apply a single rate to all particles falling within the confines of one cell. This procedure leads to computational difficulties as seen in the figure below.

The ordinate represents the local value of the scalar variable while the absissa is marked off in cellular intervals. The dashed lines represent cell-based values of the scalar while the solid line traces through the particle-based distribution. The arrows represent the increase or decrease of
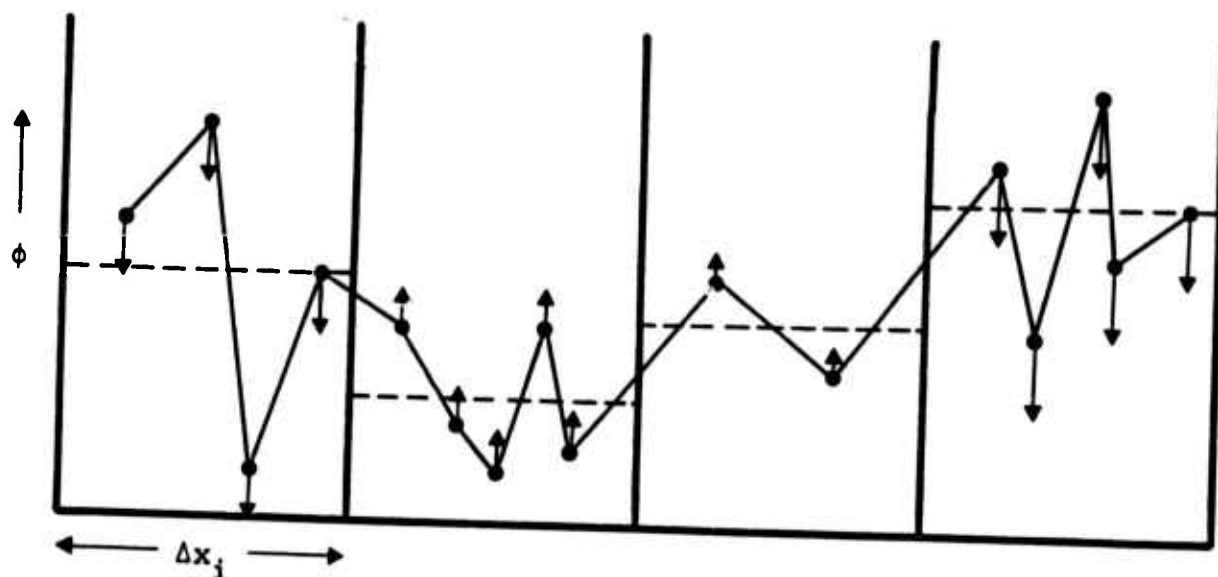
particle-based values when the net "source" rates are applied to the particles falling within each of the cells. One sees that in many instances a universal application of cell-based rates to the internal particles could result in some particle-based values going negative. As a result of this, a weighting technique was developed which drives the average particle value to what would be the new Eulerian value but also drives the particles which deviate most from the norm at the fastest rate. The effect is to smooth particle distributions such that large variations are damped and hence the Lagrangian distribution remains undistorted over longer integration times.

STEP 4 — Solution of the momentum equations.

Utilizing the boosted eddy viscosity calculated in Step 1 and the old velocity field, the subroutine MOMENT determines the resultant new field. The procedure is to first calculate the diffusion terms and store these in temporary storage. Next, the advection terms are calculated and temporarily stored. Finally, the buoyancy term is determined. A resultant tentative velocity field is then determined by applying the calculated rates for a time Δt. This velocity field is then utilized in the Poisson equation solution for the corrector potential. Finally, the resulting corrector potential is utilized to update the tentative velocities to the true new velocities as in Eq. (16). In order to achieve a second order accurate time differencing scheme in the momentum equation solution, a second iteration on the above procedure is carried out with the "old" velocity field being replaced by an average of the true old velocity field and the one just calculated. It is found that this second iteration is not very costly in terms of increasing total cycle execution times. This is not obvious since the Poisson equation solution requires

an iterative procedure in its solution. It turns out that an over-relaxed Gauss-Seidel procedure produces quite rapid convergence. Additionally, a conservative value of the Courant number also aids in rapid convergence.

STEP 5 — Move the particles.

Once the new velocity field is determined, the evaluation of the scalar advection terms is carried out. This is done by moving the particles at their local velocity for a time Δt . The local velocity is determined from two considerations: first, the time centered velocity field is used, and second, an interpolation scheme is used to determine the velocity appropriate to the position of the particle.

STEP 6 — Add or delete particles to account for sources of momentum.

Lagrangian marker particles must be added or removed as necessary from the calculation as the fluid element in which they reside is swept into or out of the computational mesh. Additionally, the discrete representation of the velocity field will sometimes result in particles being moved into internal obstacle cells accidently. These conditions are handled in two ways depending on the boundary conditions.

If the cyclic boundary option is specified, then particles swept out of the downstream side are reinserted at the upstream side and complete their movement for that time step. Any particle accidentally being moved into an obstacle region is placed at the position it occupied before the movement. This insures a constant number of particles within the fluid.

If an a priori boundary condition is specified, then particles which are swept out of the downstream side are

flagged as deleted in their respective batches. The vacant space then becomes available for the insertion of new particles generated by fluid entering on the upstream side, or from interior sources of momentum. New particles are given scalar values characteristic of the source from which they emerged. Presently, the code deletes all particles which are accidentally moved into an interior obstacle when the a priori boundary condition is specified. This tends to reduce the number of active particles slightly but produces little alteration of results when compared to the repositioning method.

All of these calculations occur in routine PMOVE which also performs the search of the particle batches for particles flagged as deleted. It then packs the batches eliminating the deleted particles and inserting any newly generated ones into the vacancies.

STEP 7 — Establish new cell-based values.

With the completion of the movement of the particles, the equations effectively have been solved at the new time. The task remains, however, to generate the new cell-based quantities. These quantities are required in the calculation of additional time steps and are desired in the editing of problem results.

The cell-based quantities are calculated in routine CENSUS which utilizes essentially the same procedure found in the MAC method.[3] Each particle is visualized to extend $\Delta x_i/N$ in each of the cellular dimensions and to have a scalar value throughout this volume equal to the value attributed to the particle. N is the cube root of the particle density. The result is to smooth the distribution and to reduce the likelihood of large discontinuities arising from particle clumping. The CENSUS routine determines the fraction each .

particle donates to each cell. These contributions are summed and normalized to yield a resultant cell-based distribution. Finally, CENSUS checks to insure that no cell is empty of particles. If any are found in this condition, then the cell value is given the averaged value of all the adjacent cells. This rarely happens in runs where the particle density is eight per cell or greater.

STEP 8 — Advance the time.

The time is now advanced so that a new cycle can begin. Several stability criteria representing different terms in the equations limit the size of the time step.

In order to avoid the problems associated with having to introduce more than one set of source particles per time step across any boundary of the domain, the stability number is restricted to values of $f < 1/N$. This allows considerable coding economy and does not restrict the allowable time steps too severely.

The stability criterion in explicit schemes requires a restriction on Δt due to diffusion. This is calculated in routine EVMAKE and has the following form:

$$\Delta t < \frac{f}{4\varepsilon} \frac{1}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}$$

where $\Delta x, \Delta y, \Delta z$ represent cellular dimensions, ε is the local eddy viscosity, and f is a stability number such that $0 \leq f \leq 2$.

Another stability criterion incorporated into STUFF reflects the stability associated with the propagation of internal waves. In order to ensure stability in this case, we require

$$\Delta t < f\Delta x/c$$

where Δx is the smallest dimension of the cell, f a stability number, and c the internal wave phase velocity given by

$$c^2 = g \frac{\lambda}{2\pi} \Delta\xi$$

where λ is the largest wavelength of the generated internal waves, g the gravitational acceleration and $\Delta\xi$ the maximum fractional variation in density across the cell interfaces. λ, in effect, is a free parameter to be chosen a priori from past experience with waves arising in fluid flow problems. Its value is not critical for most cases.

Finally, the stability condition for the advection scheme can be expressed as follows:

$$\Delta t < \frac{-\left|U^{n+1}\right| + \sqrt{(U^{n+1})^2 + 2f\Delta x \left|\frac{\partial U^n}{\partial t}\right|}}{\left|\frac{\partial U^n}{\partial t}\right|}$$

Account has been taken above of the possible increase in velocity during the anticipated time step.

The final resultant time step to be used in the next increment is chosen to be the most restrictive of the four inequalities presented above.

STEP 9 — Honor output requests.

Requests for edits of output data can assume three forms: (1) requests for a printer dump of all variables of interest including vector and scalar arrays, (2) requests for

printer plots of the velocity and scalar fields; an option is to be specified by the user of which particular variables are desired. The plots are depicted as if one were looking down from the top onto the model and successively removing layers of cells. Finally, (3) a tape dump may be made for the purposes of saving the variables for later use, such as a restart. Other output features were mentioned in the previous report.[1]

This completes a calculational cycle in STUFF3. We next discuss the initialization of a problem. The options available for starting a calculation are outlined below.

2.2.4 Initialization

The Velocity Field -- Three methods are available to the user in specifying an initial configuration. The first, which is of limited use, involves the determination of the internal momentum distribution from potential flow and specified boundary flow. This method requires the assumption that the boundary flow must be irrotational. This corresponds to an input flow uniform in z with no differential cross flow, a condition not likely to be found in atmospheric studies.

A second technique is to specify the velocity in the x-direction to be periodic and proceed as above. This has the advantage of uniquely determining the flow. A proper boundary condition is also imposed at the downstream grid face which does not introduce flow disturbances. However, this technique is of doubtful usefulness in 3-D codes since large numbers of downwind cells are required so that wrap around effects do not become significant. This approach becomes prohibitively expensive in realistic situations.

The third method appears to be the most useful for mountain wave problems. This involves the useage of the corrector potential-velocity relations expressed by Eqs. (15) and

67

(16). The procedure is to use the best guess of the ambient
field available. This will be in considerable error near
highly convoluted regions and the divergence will probably be
non-zero. This field is then utilized as a source term of
the Poisson equation to derive a corrector potential which,
when combined as in Eq. (15) with the guessed ambient field,
will result in a field which is non-divergent and preserves
the vorticity distribution of the original guess. This pro-
cedure gives realistic ambient fields, and although they are
not unique, they have represented excellent starting fields
for several test cases. This method will allow the specifi-
cation of essentially arbitrary z-dependent u-velocity pro-
files. Cross winds are also allowed.

The Scalar Fields -- Initialization of the Eulerian-
based values of the scalar fields was described in the pre-
ceeding semiannual report.[1] The particle-based values are
initialized from the above Eulerian-defined distributions.
Essentially, every particle which falls within a given cell
is initialized with the values of the Eulerian-based quantities
of density, contaminant concentration, and turbulent energy
density.

3. CONCLUSIONS

With the description of initialization procedures we have essentially completed our discussion of the Boussinesq hydro-code STUFF3. Some additional detail regarding particular topics is presented in the appendices.

STUFF3 represents one of the more advanced fluid dynamics codes in use in atmospheric research study. The mixed-mode structure and incorporation of a heuristic model of turbulence, both represent an excellent utilization of current state-of-the-art knowledge. Machine requirements are not significantly greater than previous Eulerian codes and execution times are comparable.

Future use of the code is to be discussed in the "Future Plans" section.

4. FUTURE PLANS

The above discussion essentially completes the code development work on the 3-D transient model. Its future application will be limited due to higher priority commitments, especially the wave drag parameterization study. If time and funds permit, a limited 3-D transient wave drag study using STUFF3 will be conducted.

The 2-D transient model HAIFA will be used as needed in the parameterization study to resolve questions regarding transient effects. STUFF3 will also be utilized for this purpose.

REFERENCES

1. "The Effects of Meso-Scale and Small-Scale Interactions on Global Climate," Report No. SSS-R-73-1727, Contract No. DAHC04-73-C-0003 (15 June 1973), Systems, Science and Software, La Jolla, California.

2. Chorin, A.J., "Numerical Solution of the Navier-Stokes Equations," Mathematics of Computation (October, 1968).

3. Welch, E.J., et.al., "The Mac Method: A Computing Technique for Solving Viscous Incompressible, Transient Fluid Flow Problems involving Free Surfaces," Report No. LA 3425 (November, 1965), Los Alamos Scientific Laboratory of the University of California.

APPENDIX A

THE GRID

The grid utilized by STUFF3 is pictured below. For computational efficiency and boundary condition application, the true grid is surrounded on all sides by a layer of border cells. The velocities at the interior faces of these cells can be assigned to apply various boundary conditions on the flow. Work is almost complete to incorporate outflow boundary conditions utilizing extrapolated internal values. This procedure is equivalent to the setting of the second derivative of the normal component to zero, thereby minimizing the disturbing effect of the boundary on the internal flow.

Boundary cells and interior obstacle cells are flagged with a value of zero in the grid-defining array, FULL(I,J,K). Any cell characterized by zero (meaning zero fluid) is assigned velocities associated with its interfaces at problem generation time. This allows interior obstacle cells to act as ducts through which fluid can be removed from or injected into the system, i.e., these become sources or sinks of momentum.

The particles are initially distributed within the grid at a specified density (in terms of number of particles (N) per cellular dimension). Hence, a three-dimensional problem will have N^3 particles per cell. The initial spatial distribution within the fluid filled cells is seen in Figure A-2.
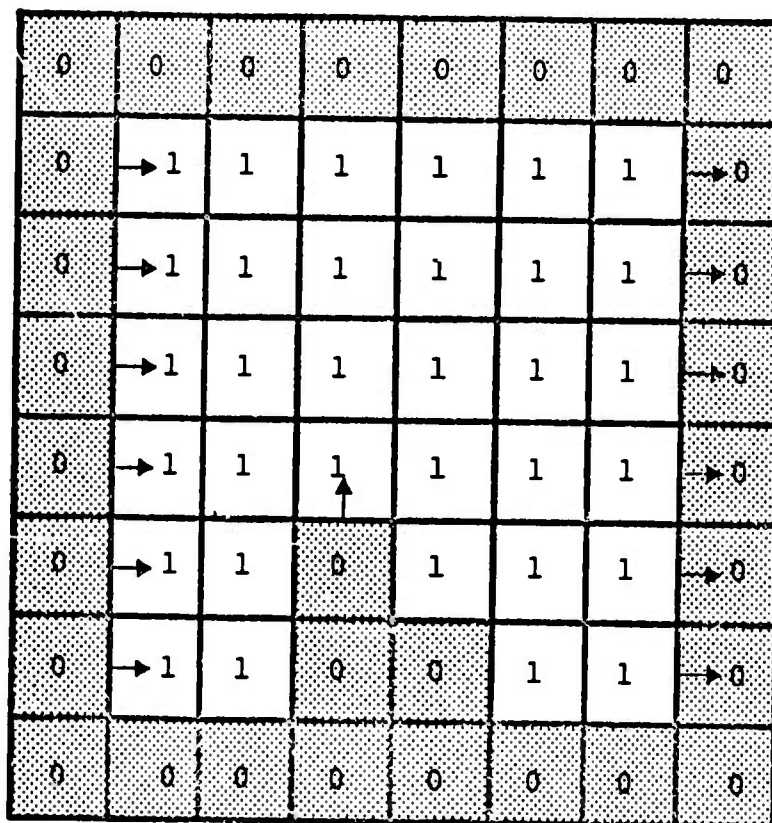
Figure A-1   Cross-section of STUFF3 grid with interior and
boundary obstacle cells. One interior cell
represents a source of momentum. Flow has been
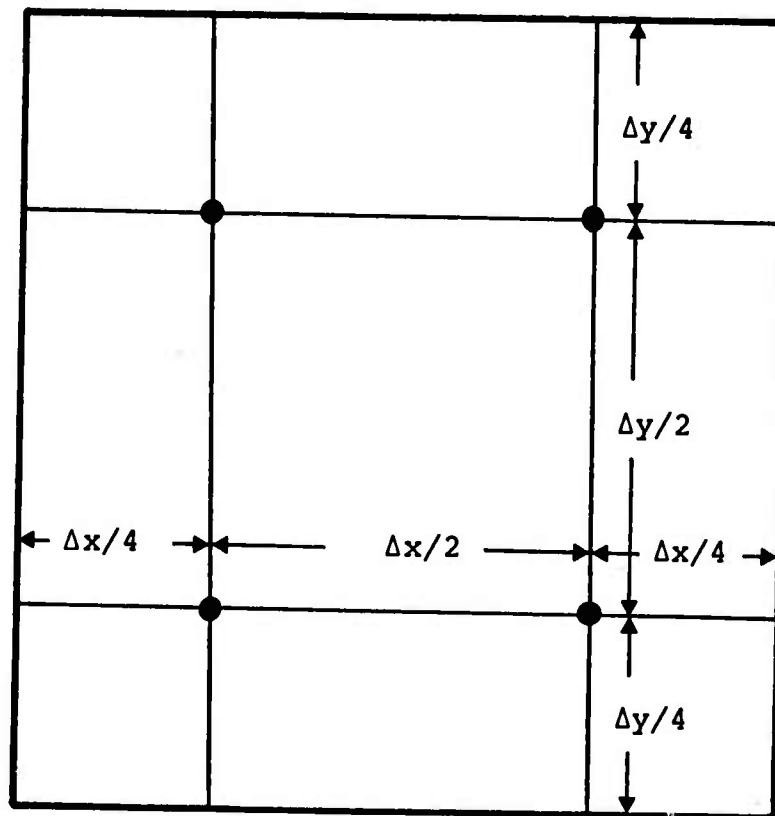specified to the right via the a priori imposed .
boundary flow.

73

Figure A-2   The distribution of particles within a given cell.
The third dimension is not shown but exhibits the
same symmetry.   N is 2.

APPENDIX B

DEFINING GRID-BASED VARIABLES

The indexing notation used to access the various scalar and vector quantities is shown for an arbitrary i,j,k cell in Figure B-1. All scalar quantities are cell-centered with i,j,k referring to the cell center coordinates. Velocities are interface quantities with the u,v,w components of a cell being defined on the $i+\frac{1}{2}$, $j+\frac{1}{2}$, and $k+\frac{1}{2}$ faces, respectively.
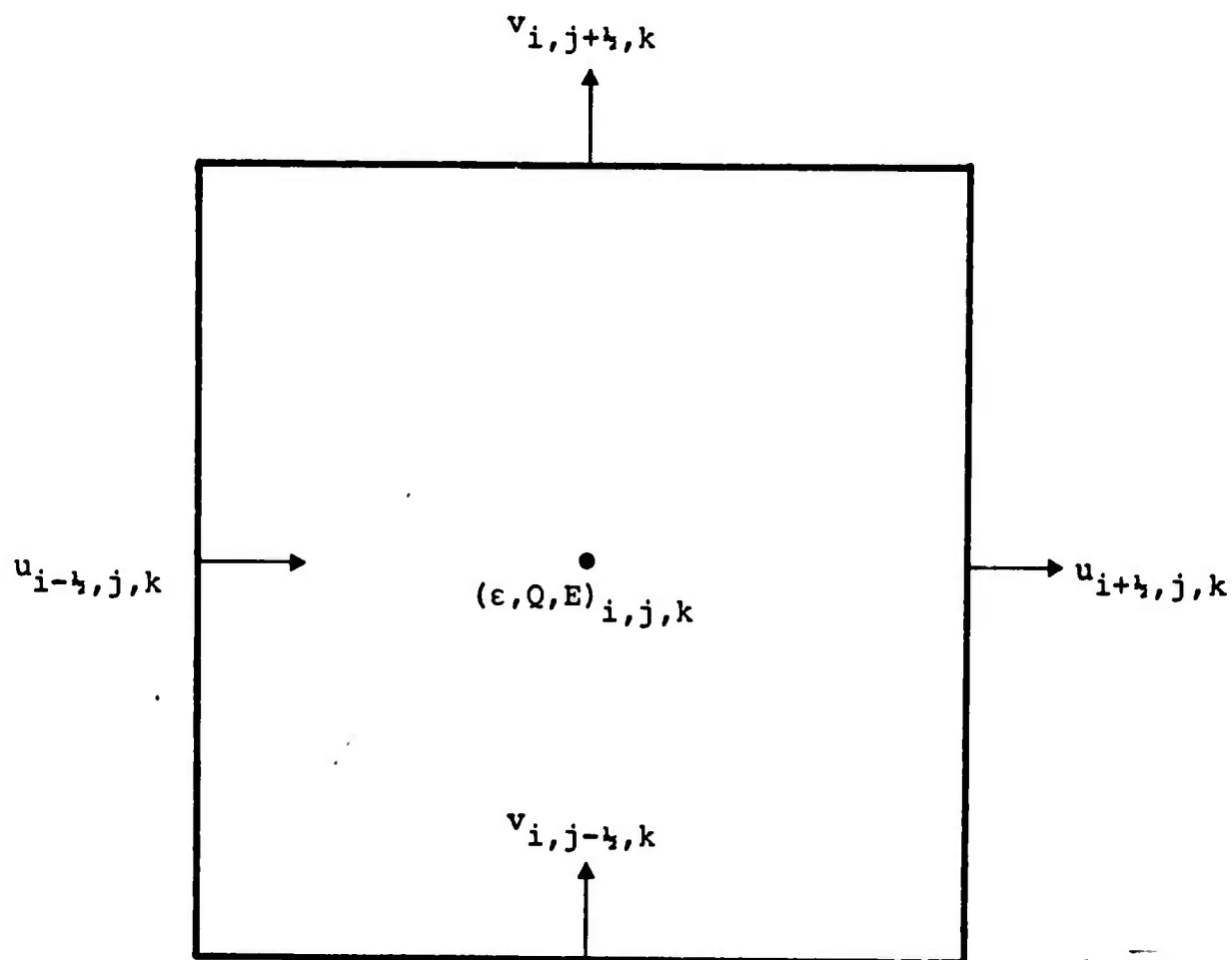
Figure B-1   Notation utilized by STUFF.   An xy cross-section
is shown taken through the cell center.   Scalars
are cell-centered quantities.   Vector components
are interface quantities.

APPENDIX C

PROCEDURE IN UPDATING PARTICLE ARRAYS
DUE TO SOURCE AND DIFFUSION TERMS

In a purely Eulerian scheme the updating of the field
variables due to diffusive contributions is straightforward.
Utilizing the standard centered difference operator, the cell-
based quantity tends with time to a value more in equilibrium
with its neighbors. Since STUFF utilizes the Eulerian-based
diffusion rates to modify the particle-based values in the
cell, it is possible, using an average rate, to force values
associated with some of the particles, negative. While a sub-
sequent census would yield a cell-based value in agreement
with the purely Eulerian method, discontinuities may develop
when these particles undergo subsequent advection. For ex-
ample, some of the negative particles might be moved into
cells in which they are the sole contributors. A subsequent
census would then yield negative cell-based quantities. In
order to avoid this problem a smoothing function was developed
which preferentially modifies particle values and reduces the
variation from the expected Eulerian value.

This function should be related to the local curvature
in the field. Also, it should be functionally dependent on
the strength of the diffusive coefficient and the time interval
over which it can act. After much experimentation, the follow-
ing form was selected:

$$Q_{PART}^{n+1} = Q_{CELL}^n + \left(Q_{PART}^n - Q_{CELL}^n\right) e^{-\Delta t S \epsilon} + \Delta t \frac{\partial Q}{\partial t}$$

where Q represents the field variable, PART and CELL refer to particle and cell-based values, respectively. ϵ represents the sum of the turbulent eddy viscosity and molecular diffusion coefficients, and S is related to the field shape by:

$$S = \frac{\sum \left(\frac{\partial Q}{\partial x_i}\right)^2}{\sum \left(\frac{\partial^2 Q}{\partial x_i^2}\right)^2}$$

where the summation range over all cells, and the derivatives are approximated by standard cell centered finite difference operators. The last term involving $\frac{\partial Q}{\partial t}$ represents the rate of change due to diffusion and sources and is determined from all based quantities at $t = t^n$ as is the S factor.

APPENDIX D

THE PARTICLE CENSUSING SCHEME

Once the particles have been moved to their final positions for the current time step, a census is carried out to determine the resultant cell-based values so that a new time step may commence. For the purposes of the census, each particle is considered to fill a finite volume, whose dimensions are given by $\Delta x_i = \Delta X_i / N$. Δx_i is the particle dimension in the ith direction, ΔX_i is the cell-dimension in the ith direction, and N is the cube root of the particle density per cell. This description is defined graphically in Figure D-1.

The calculation of the cell-based quantities now becomes a series of summations, whereby the value for a particular cell is given by:

$$\phi_{CELL} = \frac{\sum\limits_i \phi_{PART_i} \cdot dv_i}{\sum\limits_i dv_i}$$

where ϕ is the scalar variable with PART and CELL referring to particle and cell-based values, respectively. dv_i is the part of the volume of particle i falling within the boundaries of the given cell.
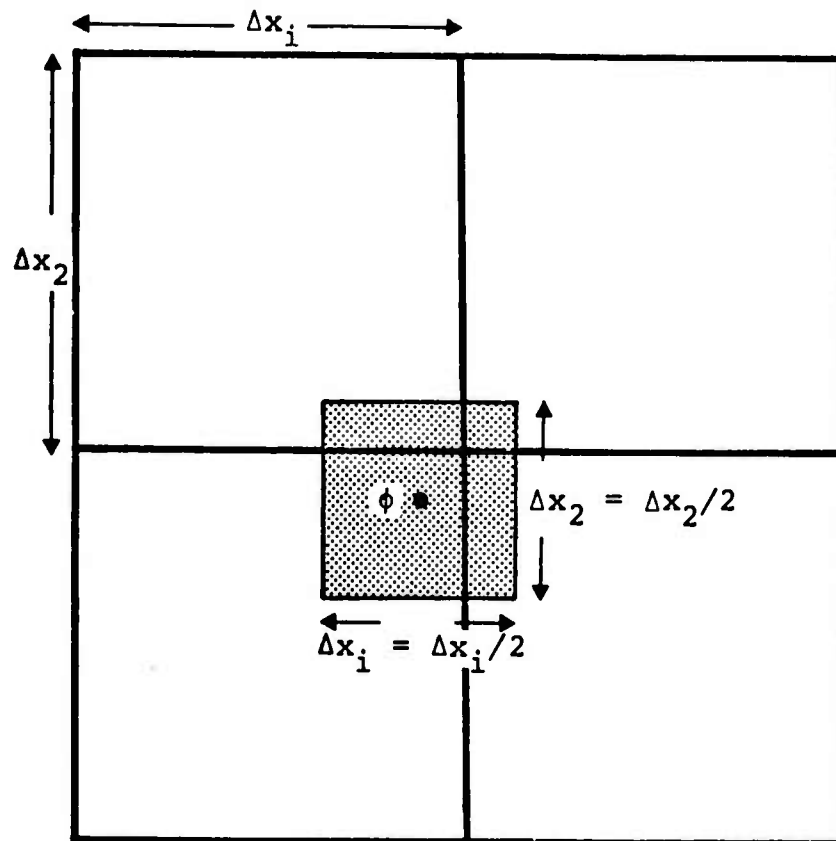
Figure D-1   A cross-section of Eulerian cells containing
a particle "cube". In this example, particle
density is 8/cell, hence $\Delta x_i = \Delta x_i/2$.

80

APPENDIX E

THE PARTICLE MOVEMENT SCHEME

The scalar advection terms are calculated through the actual moving of the particles. After the completion of this procedure a census is taken to establish the new cell-based values. This results in an evaluation of the advection terms which is free of artificial diffusion.

Some errors are introduced by this scheme, however. While they are much less significant than those found in the method they replace, they should be noted.

Each particle in the fluid should be advected by the fluid. In order for this to be the case, the instantaneous velocity of the particle must be the local velocity of the fluid. A finite difference scheme can only approximate the velocity field at each space-time point. As a result, the velocity which they experience is some interpolated value between locally defined values. Once this value has been determined the particle is moved at this velocity for the time Δt.

Two types of errors are introduced. The first results from the fact that the calculated instantaneous velocity of the particle is an interpolated one. The resulting error will be small if the Eulerian grid is of the proper resolution to

adequately resolve the problem. A second error is introduced when the particle is moved at the constant velocity for a time Δt . Actually, the local velocity is a function of t . The magnitude of this error is still significantly less than the scheme being replaced, however.

In order to insure that the scheme is second order accurate in time, the time centered velocity field is utilized for the particle movement.